AD-A115 671 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 17/2
PROTOTYPE DELNET USING THE UNIVERSAL NETWORK INTERFACE DEVICE.(U)
DEC 81 C E PAPP
UNCLASSIFIED AFIT/GE/EE/81D-46 NL

1 OF 2
ADA
115 671

UNITED STATES AIR FORCE

AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/EE/81D-46

PROTOTYPE DELNET USING THE UNIVERSAL NETWORK INTERFACE DEVICE

THESIS

AFIT/GE/EE/81D-46 Charles E. Papp
Capt USAF

AFIT/GE/EE/81D-46

PROTOTYPE DELNET
USING THE
UNIVERSAL NETWORK INTERFACE DEVICE

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Charles E. Papp, B.S.E.E.
Capt USAF
Graduate Electrical Engineering
December 1981

Approved for public release; distribution unlimited.

Preface

The purpose of this invesatigation was to construct a prototype of the Digital Engineering Laboratory Network (DELNET) using the Universal Network Interface Device (UNID). This involved completion of the prototype UNID, construction of a second UNID, and construction of the DELNET. This investigation was limited to the development and testing of the prototype DELNET, configured with two computers.

I would like to thank my thesis adviser, Dr Gary Lamont, for his assistance and encouragement during the course of this investigation. I would also like to thank my readers for their valuable comments and aid during this period. The high quality support offered by the laboratory technicians was also appreciated greatly. I would like to thank Dan Zambon and Orville Wright, in particular, for their excellent support during the construction phases of this study. Finally, I wish to acknowledge my gratitude to my wife, Bobbie, for her encouragement, understanding, and assistance during this effort.

Contents

Page

Page

List of Figures

Figure Page

Figure Page

Abstract

The objective of this investigation was to complete two prototype microcomputer based message processors and continue construction of a local network using these devices. Using a preliminary design and partially constructed hardware, the message processors were built, documented, tested, and operated. The two message processors, known as Universal Network Interface Devices (UNIDs) were then configured and operated in a prototype local network.

Chapter 1 Introduction.

Chapter 1 motivates this work by providing a survey of related network topics germane to UNID and DELNET development. Several current computer networks are surveyed, along with current technological advances, to emphasize important topics necessary for the development of local networks. These developments are then applied to an application on a typical Air Force base. Finally, a summary of the work of previous investigators is included followed by the objective, scope, approach, and overview of the work covered in this report.

1.1 Background.

Computer network theory and application, as developed over the past decade, enjoys widespread attention today. Computer networks have become an attractive alternative for increasing computational power and sharing computer resources. Undoubtedly the most successful example of computer networks is the current Defense Advanced Research Projects Agency Network (ARPANET). This massive packet-switched network not only epitomizes the future of computer networks, but also establishes network theory for subsequent new applications. (1:5-23)

Unfortunately, an international network of this size cannot be tailored to the needs of all potential network users. Using the ARPANET as a simple computer network interconnecting various mini- and microcomputers is impractical. These machines may not need the resources available through the ARPANET. A cost-effective and efficient computer network, dedicated to these smaller machines, would be more desirable. Such a computer network is commonly known as a local network (2:286-323).

The inherent benifits of local networks are being realized today through the advent of microprocessor technology. These inexpensive large-scale integrated circuits (LSI), along with a host of support LSIs, are making local networks cost-effective, efficient, and flexible. For their small size, LSI processors are becoming powerful enough to handle most of the network overhead, making the operation of the local network virtually transparent to the user.

A second fortunate development in support of local networks is research into the design of routing algorithms and the techniques of flow control. Obviously, routing algorithms and flow control have direct impact on local network applications. For example, an efficient routing algorithm insures messages reach their destinations with a minimum delay (13:213-241). Likewise, sound flow control techniques will limit congestion on the network (13:242-261). Currently, these research efforts are yielding positive results for network applications eventhough these areas of study are still in their infancy (13:212).

A third important development for local networks came with an understanding of the need for network protocols. Protocols permit one machine to communicate with another machine through rules governing the timing and formatting of the data to be exchanged (3:35). Many different protocols are now available with some movement toward standardization. The first protocols were developed as part of scientific research projects under ARPANET and the French Cyclades network (3:67-69). Commercial manufacturers have also developed various protocols. The first was IBM's development of the Binary Synchronous protocol (BSC) (2:120-121). IBM then introduced the System Network Architecture (3:70-71). More recent commercial protocols include DEC's DECNET (3:69-70) and Xerox's ETHERNET (2:292-295).

In many countries the government or private industry began offering network services to any organization wishing to subscribe (2:28). These networks are called public networks, analogous to the public telephone and utilities systems. Such widely available networks needed common agreement on some form of protocol. With this approach many users, operating dissimilar hardware, could subscribe to this service. This international standardization of protocols is currently being developed by the Comite Consultatif Internationale de Telegraphique et Telephonique (CCITT). The standards are recommended for voluntary adoption to all manufacturers and users. One of the more important CCITT recommendations, achieving widespread recognition, is Recommendation X.25 (3:71-73).

First generation local networks, when introduced on a large scale as a commercial commodity, were machine dependent. Manufacturers introduced these networks for their machines only. Interconnection of other machines through these limited networks required the development of emulators. Two examples of machine dependent local networks are IBM's SNA and DEC's DECNET.

However, within the last three years local networks with much greater flexibility became available. These networks are designed to interface many different types of machines. Two typical examples are Ungermann-Bass NET/ONE and Xerox's ETHERNET. Since the structure of local networks is important in this study, these two examples will be briefly described below. NET/ONE uses a network interface unit (NIU) built around the Z80A microprocessor (4:114-122). The transmission medium is baseband coaxial cable capable of transmission speeds up to 4 Mbps. The NIU is a significant feature of this network. A single NIU can connect to the network any one of a large number of dissimilar machines. Reconfiguration of the NIU for different machines is accompished easily by loading different software. Additionally, each NIU is isolated from the network. Should one NIU fail, the rest of the network will not be affected. Further, NET/ONE NIU's contain most network overhead, making the network operation essentially transparent to both the user and the host.

The ETHERNET is similar to NET/ONE, but not as resourceful (5). ETHERNET does not use network interface units. This requires the subscriber to house the network overhead in his machine. Instead, a transceiver interfaces each user to the network. This transceiver isolates the network from user faults as well as connecting the user to the baseband cable.

As local networks became more efficient and better understood, the US Air Force began evaluating their potential usefulness (6). For many years, telecommunications on the average AF base was limited to telephone circuits and low-speed teletype circuits. In addition, more and more base functions are being managed by digital processors and computers. Communication circuits and computers are two components needed for local networks. Adding a third component, flexible network interface devices, could make possible a base wide local network. The 1842nd EEG/EEIC technical report TR 78-5 recommends use of local networks to modernize base operation and telecommunications (6).

The local network application recommended in the report is based on the multi-ring concept. This network has a number of independent ring networks connected at specific points. Each ring may contain a processor and a series of terminals organized in support of a particular base function. For example, the base accounting and finance function might use a ring network connecting terminals and printers located in the pay roll, allotment, and comptroller offices, to the base processor used for pay roll accounting. Likewise, this functional ring might be connected to another ring network such as the consolidated base personel office (CBPO), for rapid response to personel and pay matters.

Fig 1-1, taken from the report, shows a possible collection of functionally organized ring networks on a typical AF base. Notice that this multi-ring local network contains the three components discussed above: a communications system, computers and processors, and interface devices. Implementing this network at the simplest hardware level involves connecting the user to the computer or processor through the base cable system. As seen in Fig 1-1, connecting user and machine is accomplished through use of five types of network interface devices.

These devices perform the tasks of buffering, packeting, rate changing, interfacing, and routing the data messages where required throughout the network. Type 1 and 2 nodes perform basically as concentrators and message routers, connecting user terminals to the network. Type 3, 4, and 5 nodes perform these tasks in addition to more specialized functions. Type 3 and 5 nodes interface the computer or processor, and base telecommunications cable system, satisfying their particular I/O port or channel signalling requirements. Type 4 nodes connect the local

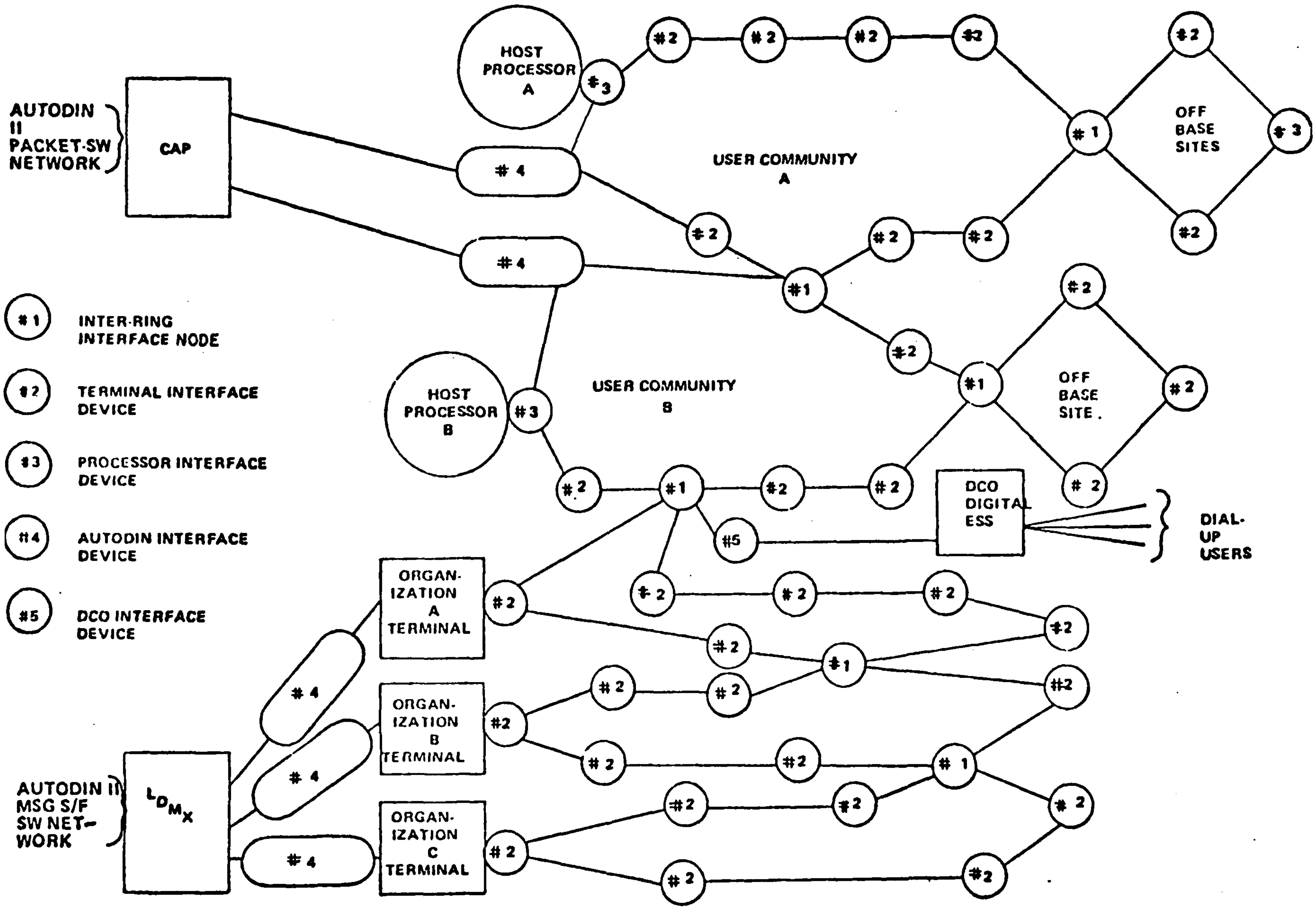


Fig. 1 Mult-Ring Base Network

network to much larger communications networks by performing all the above tasks in addition to handling the link communication protocol requirements.

Implementing this multi-loop local network depends on these five types of interface devices. Since each device performs similar tasks, one universal network interface device (UNID) could conceivably satisfy all five types of interfaces.

The subsequent development of the UNID and a local network application became the basis for numerous investigations. In 1978, Sluzevich published a preliminary design for the UNID (7). Brown, in 1979, began building a prototype UNID based on this design (1). In 1980, Baker continued building the prototype, adding modifications and refining the design (9). In 1981, Hobart began the development and design of a local computer network for AFIT's Digital Engineering Laboratory using the UNID and the loop network concepts discussed above (10). At present, a concurrent investigation by Geist continues development of software for the Digital Engineering Laboratory Network (DELNET) (11). Finally, the present investigation, completed in 1982, finished construction of the prototype UNID, built a second UNID, and implemented a prototype DELNET.

1.2 Objective.

This investigation continued development of the UNID and one network application, the DELNET. Two UNIDs were used as nodes, forming a simple local network for the Digital Engineering Laboratory following Hobart's design. The prototype network included two computers and several terminals. Subsequent operation of the network verified both the basic UNID design and the DELNET design.

1.3 Scope.

The basic hardware and software for the UNID and DELNET are necessarily constrained by the design decisions and implementations of previous investigators. This investigation remained within these limits since no substantially new hardware designs were developed. However, many details left by previous investigators were identified and implemented. These included use of a fiber optic link as the network transmission medium, expansion of UNID memory, and configuration of I/O ports. These and other hardware details were resolved, making the UNID nearly complete.

The prototype DELNET also required some hardware decisions. Connections between the computers and the UNID were specified. A few operating parameters of the network were altered, such as baud rate, I/O port configuration, and transmission media. Performance under these conditions established a measure of how well the DELNET and the UNID met original design objectives.

The UNID and DELNET required much software support.

Geist, a co-investigator, supplied this support. This investigation, electing to concentrate on hardware implementation, used his software and previously developed software.

The development of the UNID and the DELNET suffered from the lack of current documentation. This investigation updated all cable connections, schematics, lists, etc., necessary for further development. This information, along with a preliminary user's manual, is contained in the Appendicies.

1.4 Approach.

The prototype UNID was çompleted first. This included expanding UNID memory to its full limit, and completing the local I/O port card. Next, the second UNID was built, identical to the prototype. When both UNIDs were finished and tested, a simple local network was built. Several terminals were connected to each UNID and the UNIDs were connected together through a broadband communication channel. Finally, two processors in AFIT's Digital Engineering Laboratory were added to the network through the UNIDs forming the prototype DELNET. Simple tests verified the UNID and DELNET accomplished the basic design objectives published by Sluzevich and Hobart.

1.5 Overview.

This report covers all hardware aspects of the UNID and the prototype DELNET. Chapter 2 presents a summary of established requirements and design followed by a discussion of modifications implemented during this investigation. Chapter 3 details all construction for the UNID and the DELNET. Hardware testing for all components is documented in Chapter 4, including a discussion of the testing scheme. Chapter 5 summarizes the results of this investigation and recommends areas for further study. Finally, the Appendicies contain all hardware documentation and a preliminary user's manual.

Chapter 2. Requirements and Design.

This chapter summarizes the requirements and design decisions established by Sluzevich and Hobart (7,10). These decisions formed the guidelines for this investigation. Next, implementation of the prototype DELNET following these guidelines is described.

2.1 UNID Requirements Summary.

Sluzevich used the 1842 EEG report to motivate the design of the UNID. The UNID, being universal, was based on the following general concepts (7:11-13):

1. The UNID functions as a store-and-foward concentrator with message routing capability.
2. The UNID includes specialized I/O ports to handle unique communication requirements.
3. The UNID easily handles various network operating systems and communication protocols.

Using these concepts and structured anaysis techniques (SAT), Sluzevich developed UNID functional requirements (12,7:11-13). This approach developed an activity model consisting of a series of diagrams showing the UNID functions and their interrelations. Below is an outline summarizing these functional requirements.

I. Communication Interface.

   A. Signal Characteristics.
   B. Port Configuration.
   C. Protocol Software.

II. Local Information Processing.

   A. Receive local information.
   B. Store information.
   C. Process local information.
   D. Transmit information to network.

III. Network Information Processing.

   A. Receive information from network.
   B. Process information from network.
   C. Retransmit network information on network.
   D. Transmit information to local receiver.

      1. Process control information.
      2. Transmit information to subscriber.

Once these general functional requirements were stated, Sluzevich established broad UNID bounds necessary for subsequent function allocation and design phases. Below is a list of these initial system bounds. Justification is presented in (7:34-46).

UNID modularity based on circuit cards.
Software implementation of different protocols.
Synchronous, serial data rates up to 1.5 Mbps.
Minimum of one full duplex network I/O port.
RS-232C interfaces for local I/O ports.
One local I/O port for 20 mA current loop.
Interrupt controlled I/O ports.

The functional requirements allocation to UNID hardware and software was completed after establishing system bounds. At this point, three major components of the UNID were identified: local I/O, network I/O, and processor. The following lists summarizes the allocation to these components (7:48-49).

Local I/O Component (Hardware)

Recognize start of local information.
Recognize end of local information.
Recognize end of local message.
Transmit local information.

Network I/O Component (Hardware)

Recognize start of network information.
Recognize end of network information.
Change serial information to parallel.
Transmit network information.

Processor Component (Software)

Information to be transmitted.

Store outgoing local information.
Convert to network character set.
Identify ready to be processed information.
Process information to be transmitted.
Determine rout'ng.
Initialize transmitter.
Identify information as sent.
Deallocate stoage space.

Information received.

Store incoming network information.
Determine if error free.
Process information from network.
Identify as ready to be transmitted.
Identify type of message.
Remove protocol information.
Process control information.
Transmit information to local subscriber.

The information listed above resulted from the SAT activity model. Each entry corresponds to a particular diagram of the activity model (7:48-49).

2.2 DELNET Requirements Summary.

With the design of the UNID completed, Hobart designed the DELNET, a UNID based local network for AFIT's Digital Engineering Laboratory (10). Two aspects of local networks motivated his investigation. First, local networks can increase processing power without adding more computers through resource sharing. Secondly, researchers and designers need a flexible local network testbed capable of supporting network theory development.

Hobart's design was a top down development using structured analysis techniques (SAT). Design began by establishing user requirements for a development network in the Digital Engineering Laboratory. A three-part user survey established these requirements. The following is a summary of the most important requirements for the DELNET (10:19-23):

1. Ability to transfer files across the network.
2. Ability to share peripherals attached to other hosts on the network.
3. Flexibility in network configuration and operation.
4. High percentage of availability.
5. Performance monitoring capability.

These functional requirements were then used to establish DELNET system requirements, including both hardware and software. The system hardware requirements were essentially common sense approaches to network topology, hosts, nodes, and transmission mediums

(10:27-29). Topology must be flexible and easily expandable, preventing bottlenecks that limit throughput and response time. Host computers, regardless of their sophistication, should be added easily to the network. Computers with the most powerful and/or popular peripherals should be included. The nodes should not degrade host performance, keeping the DELNET nearly transparent to the user. Nodes must reconfigure easily, accomodate different topologies and protocols, and meet network throughput. The transmission mediums must support data rates based on network throughput, response time, and bit error rate requirements. Finally, the DELNET should include a fiber optic link for research uses. System software requirements were more rigorously developed using several structured analysis techniques (10:29-30). Since this investigation is limited to DELNET hardware components, the reader is refered to Hobart's investigation and Geist's investigation for a complete development of DELNET software (10,11).

2.3 DELNET Design Summary.

Hobart's DELNET hardware design specified topology, hosts, nodes, and transmission mediums. The topology selected is shown in Fig 2-1. The basic ring architecture is a simple version of the local network proposed in the 1842 EEG report (6). The star topology for the local side of the network was chosen for three reasons (10:85-86). This arrangement decreases the number of nodes required,

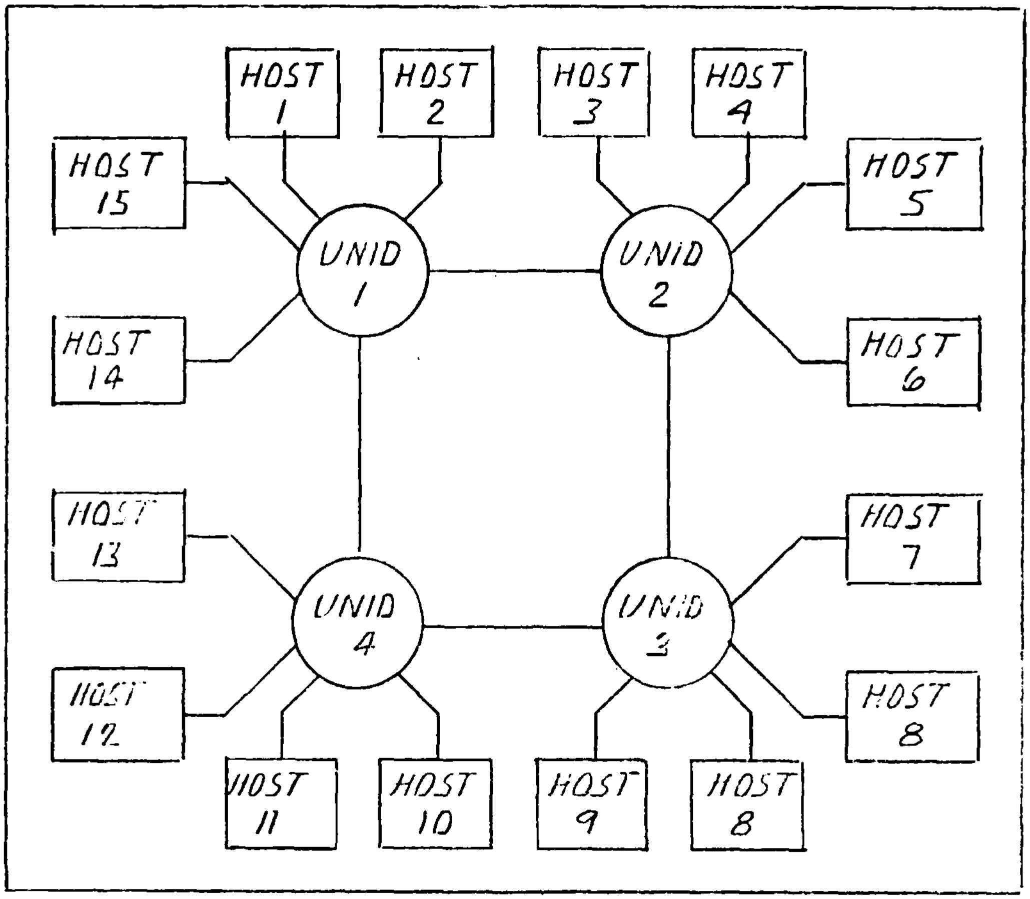


Fig 2-1 DELNET Topology.

making each node more cost effective. The topology provides a practical interface for both the simplest and most complex computers in the laboratory. Finally, the hosts that interact with each other the most can be grouped at the same node, reducing the network traffic.

Hobart specified three hosts for the initial implementation (10:86-87). These machines represent a minimum collection that would sufficiently exercise network protocol while providing the maximum resource sharing

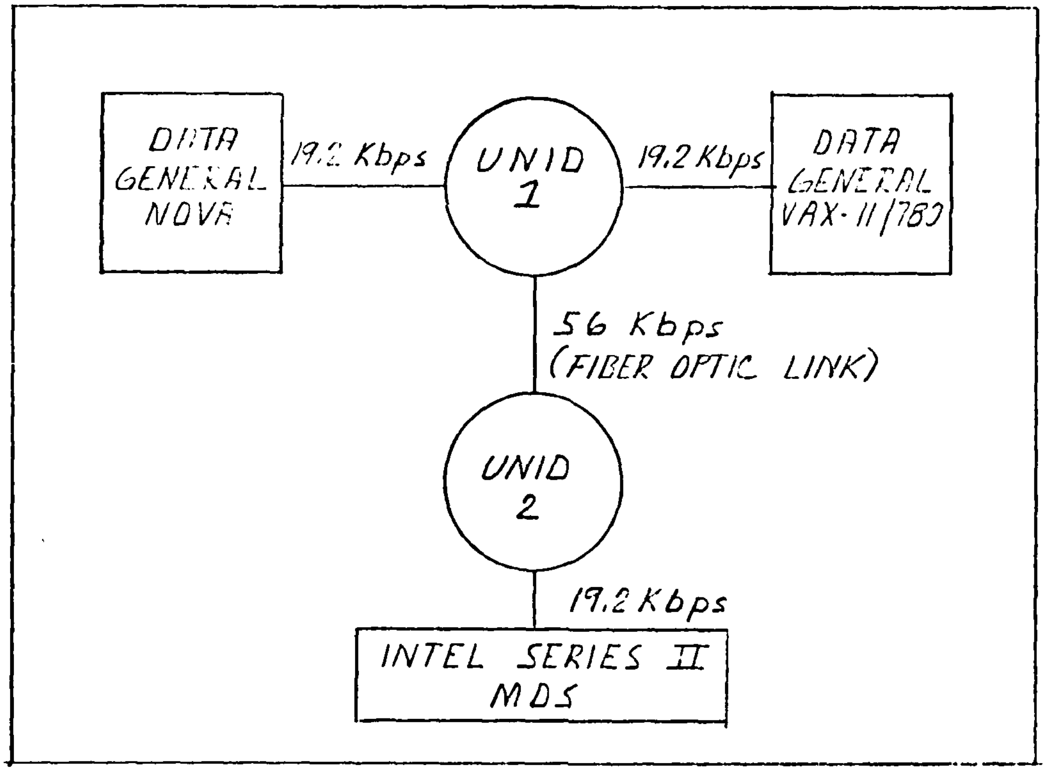


Fig 2-2 Proposed DELNET Configuration.

capability. The Vax-11/780 was chosen for its processing power and sophistication, and its capability to transfer files to and from the Cyber 750. The Intel Series II MDS was chosen for its simplicity, high usage rate, and available data base management system. The third, the Data General Nova, was chosen for its high usage rate, its link to the Data General Eclipse, and its timesharing port to the Cyber 750.

The node specified for the DELNET is the UNID (10:87-88). Currently available commercial nodes and networks are too expensive and not flexible enough for DELNET requirements. The UNID, however, offered several advantages. Complete documentation on development and design is available and non-proprietary. The UNID can be used in many topologies. The use of two Z80 processors permits parallel processing, minimizing throughput and response time delays. Finally, a high level language, PL/Z, and a software development system, MCZ1/25, are in place and available for further development.

The transmission mediums Hobart selected for the DELNET include both local and network channels (10:88-89). All local channels will use twisted pairs. One network link between two UNIDs will use a fiber optic communication channel.

Fig 2-2 shows the initial DELNET implementation with all the components discussed above.

2.4 UNID Design Implementation.

Before a useable form of the DELNET could be built, at least two operational UNIDs were needed. These UNIDs, as shown in Fig 2-2, were used to form a least cost minimum spanning tree (MST) network. In this case, the MST for a simple ring network degenerates to a single link connecting two nodes (13:171-174). Of course, more complex ring networks can be built by adding more links and nodes, as shown in Figs 2-1 and 1-1. But, the MST approach requires a minimum of hardware to build a useable network.

In Baker's investigation, one prototype UNID was built. It, however, did not meet the original design requirements. Specifically, UNID memory was expanded to 64K bytes and two additional local I/O ports were added. A second UNID was constructed, identical to the prototype. During construction of both UNIDs, a number of minor modifications were made. The details of construction are presented in Chapter 3.

After both UNIDs were completed, each major functional component was tested for correct operation. The major components are local I/O, network I/O, and processor, as described in Sec 2.1. The operating software was provided by Geist (11). Testing was conducted from the bottom up, beginning with the individual circuit cards, until the entire UNID was checked. The complete UNID testing scheme is discussed in Chapter 4.

The prototype DELNET was formed next, using the two UNIDs and the Digital Engineering Laboratory computers. A series of different configurations were used to test different UNID capabilities. The first configuration connected ATM-3 terminals to the link. The terminals were then used to communicate back and forth across the network. These operations tested the functions of the ports and processors. Later, two computers were added to the link. In these tests, the computers communicated to each other, testing the programmable parameters for the local ports and the UNID routing procedures. Configuring the DELNET for each test was simple since all local ports were RS-232C compatible. The network ports used the RS-422/423 standard across a commercial fiber optic communication link. Complete DELNET construction details are contained in Chapter 3, Sec 3-9.

After the DELNET was functioning, a series of tests were used verifying correct network operation. The operating and protocol software was, again, provided by Geist. This series of DELNET tests were essentially software performance measures and are discussed in Geist's report (11).

2.5 Documentation.

This onging investigation relies on proper documentation to provide a clear understanding of past work and future needs. Thus, an important part of this report was complete documentation of hardware, current to this date. The documentation includes various schematics, layout charts, wirewrap lists, and configuration diagrams. This information is found in the Appendicies.

Also included in the Appendicies is a preliminary user's manual containing additional information. Each circuit card is broken down to functional areas and the associated circuit operations are described. In addition, start-up procedures, loading, bootstraps, and configuration options are included. Also included are software operating systems and conventions.

2.6 Summary.

This chapter summarized the requirements and designs for the UNID and the DELNET. These represent the constraints placed on this investigation. These summaries are followed by UNID and DELNET design implementations and evaluations completed during this investigations.

Chapter 3. UNID and DELNET Construction.

This chapter details construction and modifications of the two UNIDs and the DELNET completed during this investigation. Component identification, power supplies, motherboards, and the six circuit cards for each UNID are discussed. Local and network communication interface circuits are also described. Significant alterations to the Shared Memory Cards and System Memory Cards were required and are documented in this chapter. Next, the prototype DELNET is described, explaining the configurations, computers, and interfaces used in the initial implementation.

3.1 Component Identification.

The major components of each UNID are the card cage, the motherboard, the local processor card, the network processor card, the shared memory card, the system memory card, the local card, and the network card. Each component is identified by name, serial number, and card slot, as applicable, printed on the upper righthand portion of the component. See the example below:

Network Processor 2, J2

The card named is the Network Processor Card. The 2 indicates it is part of the second UNID. J2 is the card slot in the second UNID where it is installed.

For this investigation UNID 1 and UNID 2 are considered identical. All circuit cards in one UNID may be interchanged with their corresponding cards in the other UNID. Fig A-1, Appendix A, shows the motherboard layout of all circuit card edge connectors and cable connectors for both UNIDs.

3.2 Motherboard Construction.

The motherboard used for each UNID is Zilog's commercial wirewrap backplane with nine card capacity. Each motherboard is housed in an anodized aluminum card cage also supplied by Zilog. The majority of Motherboard 1's wiring was completed during Brown's and Baker's investigations (8,9). The remainder of Motherboard 1 and all of Motherboard 2 were completed during this investigation. The following four sections discuss power distribution and wiring modifications for each motherboard. Motherboard 1 and 2 function identically and are completely interchangeable.

3.2.1 Motherboard Power Distribution. Each UNID requires four DC voltages for operation. These voltages are connected to the motherboard by a five-post terminal strip. Each voltage is identified on the edge-connector side of the motherboard. Three voltages, +12VDC, -12VDC, and -5VDC are distributed across the motherboard by adding wirewrap connections as needed. The fourth voltage, +5VDC, and chassis ground are distributed across the motherboard internally. Two conducting layers are deposited onto each motherboard. The top layer is the +5VDC plate and the bottom is chassis ground. Each of the nine female card edge connectors are attached to +5VDC and ground via these two layers. Pins 1, 2, 3, 59, 60, and 61 on each card connector are attached to +5VDC. Pins 62, 63, 64, 120, 121, and 122 are attached to chassis ground.

A problem with this two-layer power distribution existed on both motherboards. The feedthroughs for chassis ground (pins 62, 63, 64, 120, 121, and 122) on all Motherboard 1 card-edge connectors were not soldered to the ground layer when the board was received. This was corrected by Baker during his investigation by hand soldering these pins to the ground layer. A particular technique had to be used to insure good electrical contact between these pins and the ground layer. Bare 30-gage wire was wrapped around each pin to be grounded. After wrapping the wire, the small coil formed was then pushed on the post until it made contact with the ground-layer feedthrough at the base of the pin. The wire, pin, and feedthrough were then soldered together. The bare wirewrap helped solder flow into the feedthrough at the base of the pin. This technique proved satifactory for this type of problem.

Motherboard 2 had a similar problem. In this case the +5VDC terminal post on the input power terminal strip was not connected to the +5VDC layer. Evidently the internal card connection had somehow broken. A soldering iron was used to reheat the terminal post in an attempt to resolder the break. This did not correct the problem. The next attempted repair tried to attach the +5VDC to a series of feedthroughs in the left rear corner of the card. Unfortunately, the heat from the soldering iron damaged the insulation between the +5VDC and the ground layers shorting the two together. Since no repair was possible on such damage, the shorted portions were isolated from the rest of the motherboard by cutting through the two layers. The internal break at the terminal post was finally corrected by installing feedthrough pins at the +5VDC and Chassis Ground terminals. 22-gage wire was then used, connecting these two terminal posts to their corresponding wirewrap pins on card-edge connectors J7, J8, and J9. The remainder of the card-edge connectors received their power through these three connectors and the conductive layers.

3.2.2 Power Supply Requirements. Each UNID, operating with its full complement of six cards, requires approximately 8.25 amps from the +5VDC power supply. This value was measured with an ammeter connected in series with the supply and is the result of bias voltages and leakage currents associated with standard transistor-transistor logic (TTL). The current requirements for the other three supplies were measured at less than 400 mA, total. Of these, the +12VDC supply provides about 250 mA, due mainly to the leakage currents of the dynamic memory ICs used (14). The two remaining supplies are providing less than 100 mA each (14,15).

Eventually, each UNID will be powered by a switching power supply, the Power Mate ESM-200-4001, rated at 10 amps (16). One power supply was received; however, its +5VDC section was defective. The supply was returned and a replacement was ordered. In the mean time each UNID was powered by two laboratory power supplies. The Trygon Electronics' Model M60-10-0V power supply was used for +5VDC since it was the only available supply with enough current capacity. The three-section Powertec Model 6C3000 power supply was used to supply the other three voltages.

The laboratory power supplies provided good regulation with essentially noise free power. An important feature of these two power supplies is their isolated DC grounds. The isolation prevents AC ripple and similar coupling interference reaching the UNID. All DC grounds were strapped together at the power supplies to further limit power supply interference. If different power supplies are substituted, care must be used to insure they are noise free and have isolated DC grounds.

Each UNID power cable was equipped with quick disconnect Johnson connectors. Each connector is wired identically so that either UNID will operate from a given set of power supplies. This helps during troubleshooting when one UNID is replaced by the other. Power supply wiring remains in place when UNIDs are interchanged.

UNID current requirements vary considerably depending on the number of circuit cards installed. Quite often isolating hardware faults required operating a UNID with less than its full complement of cards. Each time this was done all four power supplies required readjustment. Failure to do this will subject the installed cards to voltages outside specified tolerances. This will affect long term IC reliability and may even permanently damage ICs (17). This difficulty will be corrected by the self-regulating ESM-200-4001 power supplies when they are installed.

3.2.3 Removal of Arbiter Wiring. During Brown's investigation, a circuit card was built and installed in the J4 card cage slot (8:7-18). This card was the Dual Processor Card designed to let the two processor cards access shared memory. An improved design was later implemented by Baker, eliminating the need for this card (9:33-34). During the current investigation, the Dual Processor Card was dismantled and the blank wirewrap board was used for Network Card 2. Additionally, much of Motherboard 1 and 2 arbiter wiring to the J4 edge connector was also removed. However, not all of the wiring was removed. Consequently, the remaining wiring on the J4 edge connector is not used and not documented in this investigation.

3.2.4 Motherboard Wiring for System Memory Card. This investigation implemented Baker's design for system memory

(9:40-43). Construction of the system memory cards will be discussed in Sec 3.7. All motherboard wiring for these cards was added during this investigation. The system memory card required two independent sets of signals since the card contained two independent blocks of memory, one controlled by the local processor card and the other controlled by the network processor card. Each set of signals included 16 address lines, 8 data lines, and 4 control signal lines. All of these lines were added to the system memory card-edge connector, J1. Fig B-1 in Appendix B is the schematic containing this added wiring.

3.3 Shared Memory Card Construction.

The construction of the two 32K byte shared memory cards is described in the following sections. Particular attention is given to a discrepancy in the manufacturing of circuit card blanks used for each shared memory card. The organization of shared memory is also described. Shared Memory Card 1 and 2 function identcially and are completely interchangeable.

3.3.1 Reversed Card Images. Shared Memory Card 1 was partially completed by Baker during his investigation (9:33-40). This card and Shared Memory Card 2 were completed during the current investigation. The blank circuit cards used were constructed by Baker when he discovered he needed more IC capacity per card than was available on the Z80 commercial blank cards (9:32-33).

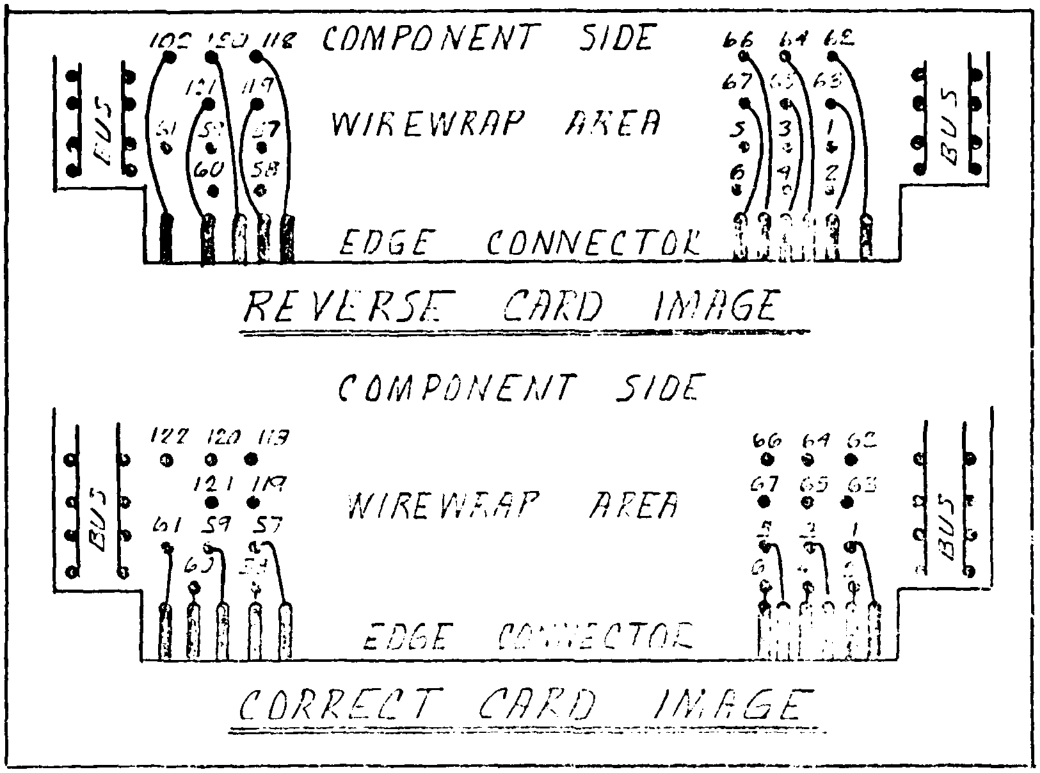


Fig 3-1 Correct and Reversed Card Images.

Eventhough these two memory cards function identically and are interchangeable, a significant discrepancy exists between these two cards.

Baker built seven wirewrap card blanks for eventual use in the UNIDs. Unfortunaterly, some of the cards were manufactured incorrectly. The photographic negative used to construct these cards was reversed during manufacturing. This gave the cards a reversed image of the correctly photographed boards. This reversed image affected the edge connector pin pattern at the bottom of the circuit card. Fig 3-1 illustrates the difference between the correct board and the incorrect reversed image board. Notice that the contacts connected to pins 1-61 on the correct card are connected to pins 62-122 on the reversed image card. Care must be used when tracing wiring on these boards.

Shared Memory Board 1 was constructed on the correct board. But, Shared Memory Card 2 was constructed on the reversed image board. This discrepancy was not detected until after completion of both cards. Careful attention to this discrepancy will eliminate future confusion. The only difference between these two cards is the way the edge connectors are wired. The schematic shown in Appendix B, Fig B-2 reflects the card-edge wiring for Shared Memory Card 1. The pin numbers in parenthesis next to these numbers are the card edge wiring for Shared Memory 2.

3.3.2 Construction of the Shared Memory Cards.

As mentioned above, both shared memory cards were constructed on specially designed wirewrap boards. Although each board contained enough space for 70 18-pin IC sockets, each shared memory board required a minimum of 77 chips. An extension for each board was made and bonded to the top edge. The IC sockets were then installed and glued into place. Fig A-2 in Appendix A shows the layout for these two memory cards.

Shared memory, built from static RAM ICs, requires +5VDC and DC ground (18). Each verticle edge of the card has two buses with wirewrap pins soldered into them. These are the power buses and are an integral part of the grided power matrix and decoupling arrangement. Fig 3-2 shows the power distribution scheme followed for the static RAM ICs on

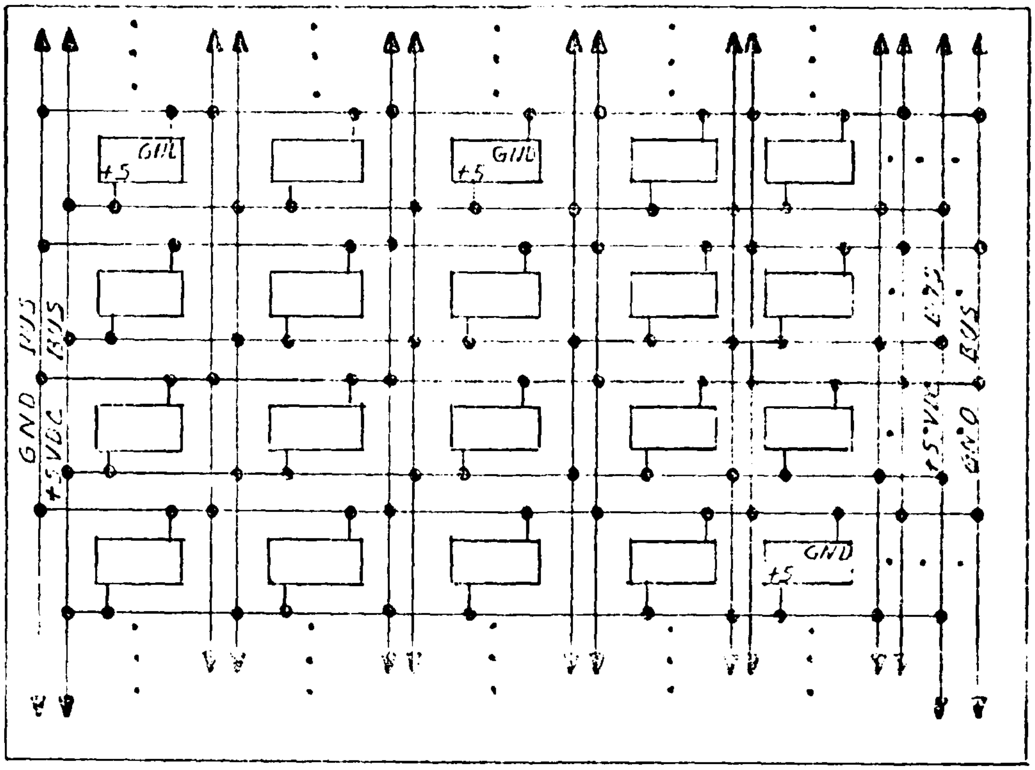


Fig 3-2 Power Distribution Grid.

the shared memory cards. Essentially, each memory chip is wired to the power bus both vertically and horozontally with a single capacitor across alternate IC power pins. This scheme follows a heuristic static memory design established by Intel, Inc and Texas Instruments, Inc (19:7-16,20:85-93). This design for the matrix arangement is discussed in Sec 3.7.3.

3.3.3 Shared Memory Organization. The bottom two IC rows on each card contain all the control circuitry necessary for memory operation. The remaining rows are composed of static RAM chips. The organization of these memories are shown in

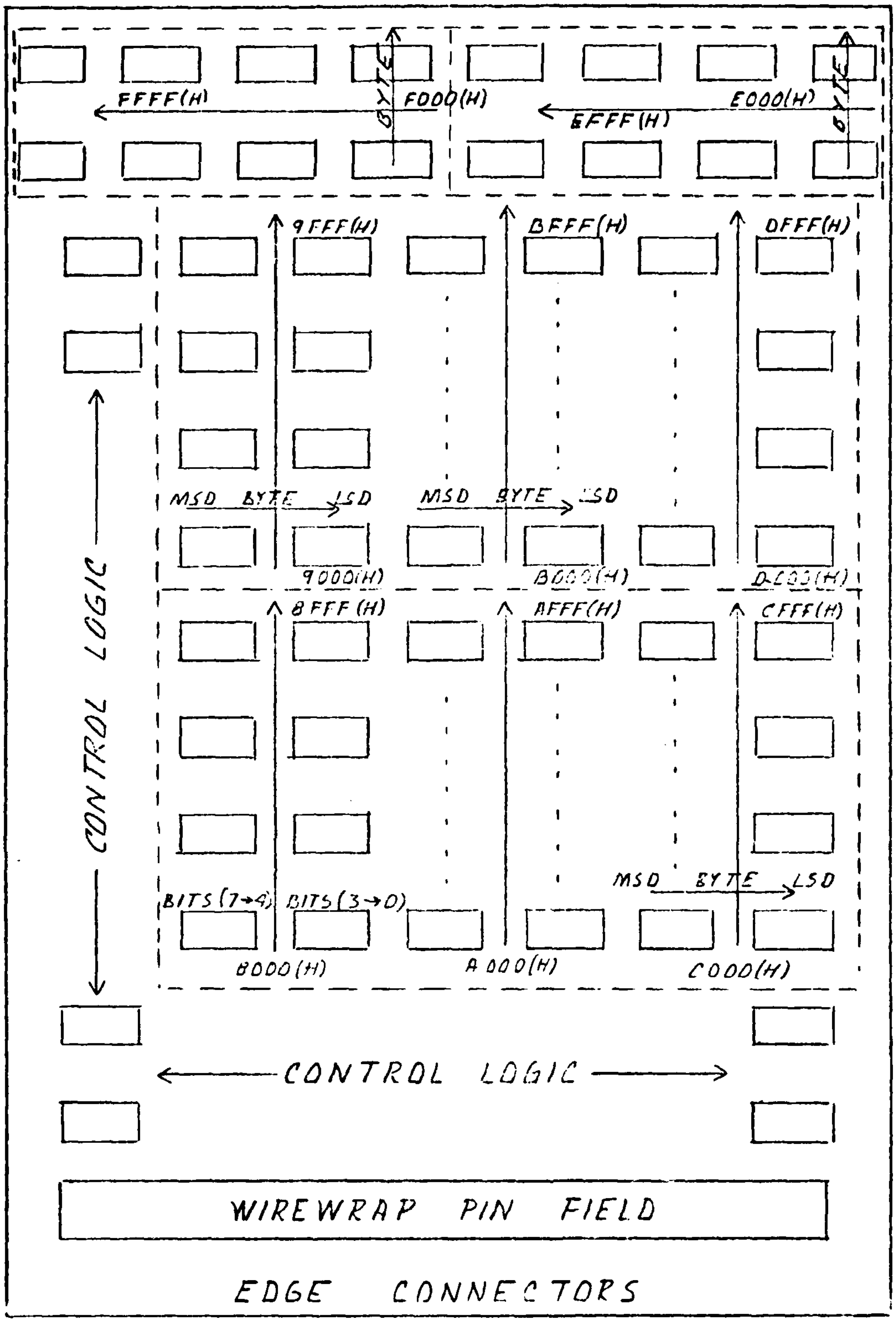


Fig 3-3 Shared Memory Organization.

Fig 3-3. The interconnecting wiring on the motherboards for these two memory cards is shown in Fig B-1, Appendix B.

3.4 Local Processor Modifications.

Each UNID uses a single Z80 Microcomputer Board (MCB) as a processor to control the local side of the UNID (7). These MCBs are extreemly flexible, capable of a wide range of applications (15). This section describes how these MCBs were configured for use as local processor cards. Local Processor Card 1 and 2 function identically and are completely interchangeable.

3.4.1 Memory Page Placement. Each MCB has enough space on its card for 4K bytes of ROM and 4K bytes of RAM. In addition, each MCB can address up to 64K bytes of memory. Both MCBs begin ROM memory at 0000(H) and RAM memory at 1000(H). This automatically places the 24K bytes on the system memory from 2000(H) to 7FFF(H) and the 32K bytes on the shared memory card from 8000(H) to FFFF(H). The monitor used with the local processor card resides in ROM, but does not exceed 2K bytes. Thus, the additional 2K bytes of ROM are not currently used.

Placing memory at the proper boundaries for each processor was accomplished by configuring a set of jumper plugs. A complete description of memory page placement and jumpering instructions is given in (15).

3.4.2 Local Processor Card Power Requirements. The local processor cards were received with Zilog's 1K monitor installed. These monitors are described in Zilog's literature (21). Baker wrote a more flexible monitor and stored it in 2K bytes of EPROM installed on Local Processor Card 1. During the current investigation the monitor was duplicated and installed in Local Processor 2. (Further information on programming these EPROMs is contained in Appendix C.) The Zilog monitors were replaced with the EPROM monitors. Replacing ROMs with EPROMs on the MCBs required two more voltages for proper operation, -5VDC and +12VDC. These voltages were supplied to the MCBs through the motherboards and certain jumpering arrangements. A descrition of EPROM power and jumpering is found in the MCB Hardware User's Manual (15).

3.4.3 Local Processor Options.

Each MCB has a series of options selectable at the motherboard card-edge connector (15). The options selected for both processor cards are as follows: high speed serial transmission, select on-card ROM, select on-card RAM, select CTC serial clock for communication interface, and disable serial interrupt to communication interface.

3.4.4 Local Processor Monitor.

The local processor monitors used for this investigation are unchanged from the monitors written by Baker. A complete listing of this monitor in Z80 Op Code is included in Appendix D. An assembly language listing is also available.

The primary functions of the local processor monitor are loading, executing, and debugging software. The available commands are: Fill, Load, Move, Next, Display, and Set. A complete command language grammar is contained in Appendix E. Further command language information can be found in references (9) and (21).

3.5 Network Processor Card Modifications.

Many of the modifications done on the local processor cards are also done on the network processor cards. In addition, the network processor card clock is disabled and a slightly different monitor is used. The following sections will describe these modifications. Network Processor Card 1 and 2 function identically and are completely interchangeable.

3.5.1 Memory Boundary Placement. A second Z80 MCB is used in each UNID to control the UNID's network side. The on-card memory boundaries for each network processor card are the same as those for the local processor cards given in Sec 3.4.1.

3.5.2 Network Processor Card Power Requirements. The Z80 MCB ROM monitors were removed and replaced with the network processor monitors written by Baker. These EPROMs have the same power requirements as the local processor monitors described in Sec 3.4.2. The required network processor power is supplied in the same manner as the local processor power.

3.5.3 Network Processor Card Modifications. Brown discovered the local and network processor cards would operate together without interfering if the network processor clock was disabled, using instead the inverted clock from the local processor card (8:64-65). Disconnecting pins 11 and 12 on A38 disabled the network processor clock. The local processor clock was then fed to the network processor through the motherboard and a jumper wire installed on the network card between edge connector pin 39 and pin 12 on A38. Since the network processor clocks were disabled, this processor card will not operate unless the local processor card is installed.

3.5.4 Network Processor Card Options. The options selected at the card-edge connector for the network processor card are the same as those selected for the local processor card. See Sec 3.4.3.

3.5.5. Network Processor Card Monitor. The network processor monitor used in this investigation is unchanged from the monitor written by Baker (9:11-17). A complete listing of this monitor in Z80 Op Code is included in Appendix D. An assembly language listing is also available.

The network processor monitor includes the same commands available in the local processor monitor described in Sec 3.4.4 except for the Load command. This command was deleted since there was no need to load software through the network side of the UNID (9:11-17). Except for the Load command, the command language grammar for the network processor monitor is identical to the grammar for the local processor monitor. The command language grammar is contained in Appendix E.

3.6 Local Card Construction.

The construction features of each local card will be described next. Local Card 1 was constructed on a commercial Z80 wirewrap card with certain modifications. Local Card 2 was constructed on one of Baker's manufactured cards. This card has the correct image. In addition, the required RS-232C interfacing for each local I/O port is discussed. Local Card 1 and 2 function identically and are completely interchangeable.

3.6.1 Local Card 1 Completion. Local Card 1 was completed during this investigation, adding two additional I/O ports. This brought the number of local card ports up to its full complement of four. Completing this card required modifications to accommodate one additional CTC and two additional USARTs. The IC layout of this card is shown in Fig A-3a, Appendix A.

Local Card 1 was built on a Z80 commercial card blank. This card was modified to accept the additional ICs by using two verticle rows of 16-pin IC sockets as though they were a single row of 24-pin sockets. Pins that were connected internally to the power and ground layers were isolated.

Pins were also installed in the rows where needed. The width of these modified rows was slightly greater than the width of the pin rows on the CTC and USARTs. The pins on these ICs were spread slightly until they fit into the board, eliminating the need for special sockets. All wiring for the two ports was added to the board. The complete schematic for Local Card 1 is shown in Fig B-3, Appendix B.

3.6.2 Local Card 2 Construction. Local Card 2 was consturcted using the same schematic shown in Appendix B, Fig B-3. Layout for this card is shown in Appendix A, Fig A-3b. The blank circuit card used was one of the cards developed by Baker (9:32-33). This particular card does not have the reversed image problem of the shared memory cards described in Sec 3.3.1.

3.6.3 RS-232C Port Configurations. Each local card will accommodate four independent RS-232C communication channels. Jumper plugs permit easy configuration of these channels as either Data Communications Equipment (DCE), Data Terminal Equipment (DTE), or a hybrid of these two. For further information on I/O port configurations refer to (22:8.1-8.5).

A modification to the original design of these ports permitted easier configuration. In Brown's design, each port had the DSR and CTS input signals hardwired together (8:34-36). This was acceptable for DTE operation; but, it prevented DCE or other hybrid configuration. The

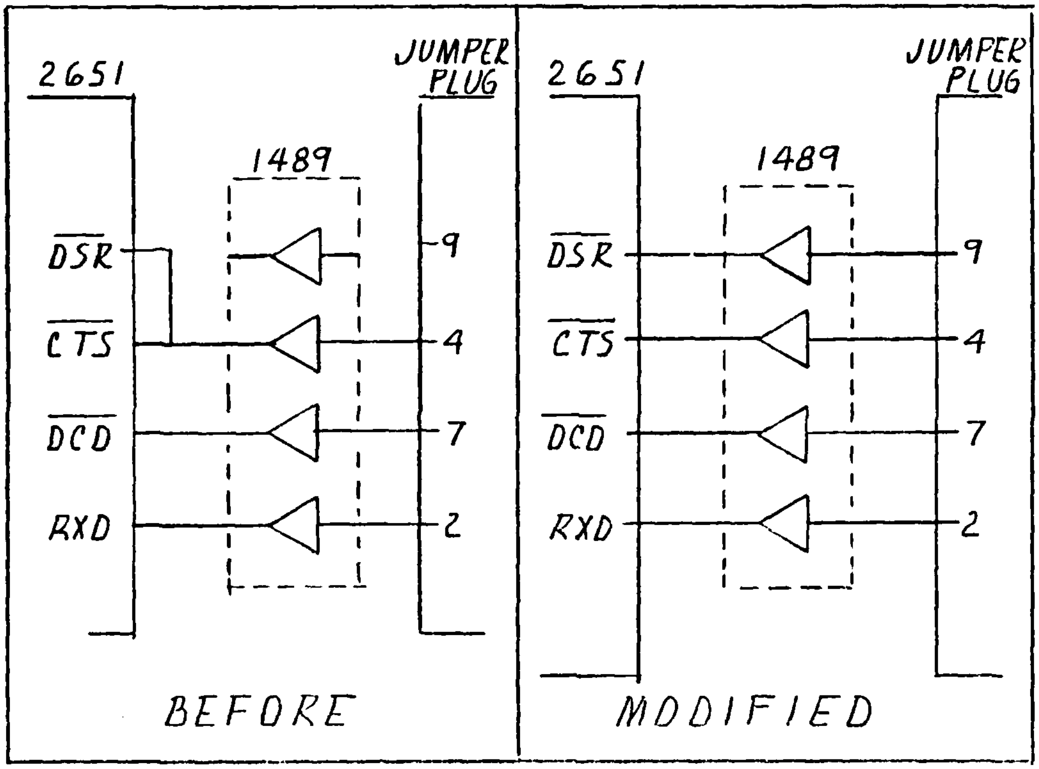


Fig 3-4 Local Card Line Receiver Modification.

modification separated these two inputs so that other configurations are now possible. Fig 3-4 shows the original configuration and the modification. The local card schematic in Appendix B, Fig B-3, reflects the modified configuration.

As noted above, the port configurations are selected by using jumper plugs in sockets J3, J7, J56, and J58 on Local Card 1 and jumper plugs in sockets J56, J58, J61, and J63 on Local Card 2. The jumper configurations for DTE and DCE operation are shown in Fig 3-5.

3.7 System Memory Construction.

The system memory cards were constructed during this

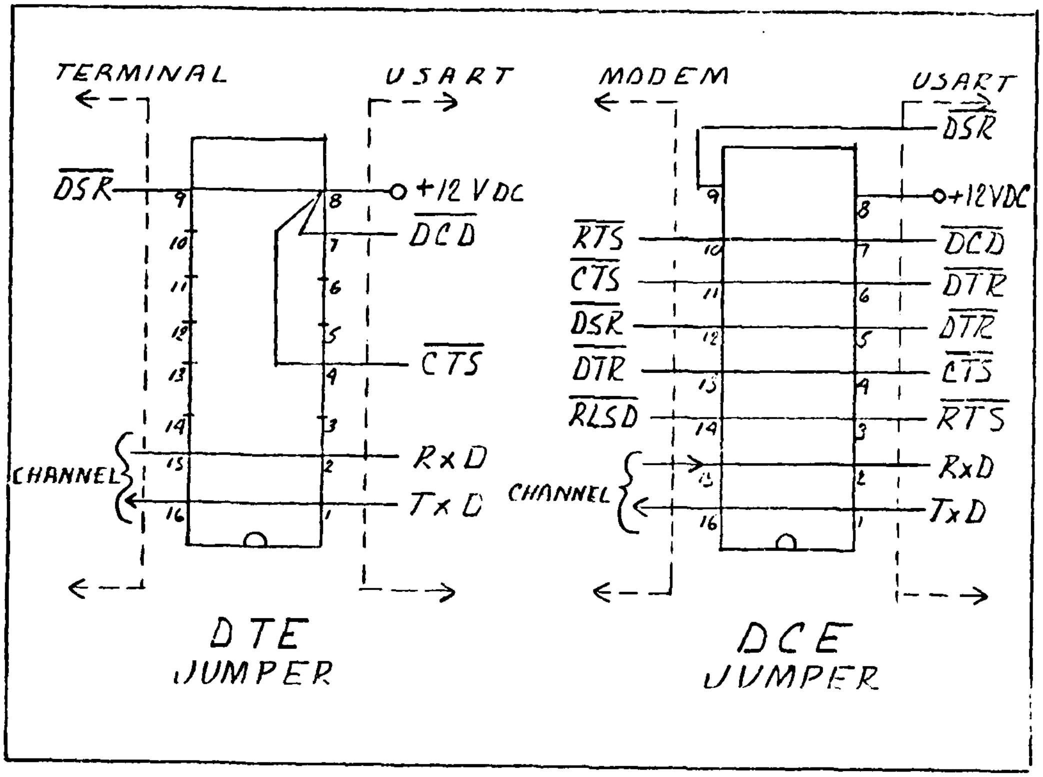


Fig 3-5 DTE and DCE Jumper Configuration.

investigation following Baker's design (9:40-43). Both circuits were constructed on the cards manufactured by Baker. One card is a reversed image card. The following sections describe the construction. Special consideration is given to the problem of power distribution and noise in dynamic RAMs. System Memory Card 1 and 2 function identically and are completely interchangeable.

3.7.1 Reversed Image Cards. The system memory cards built for the UNIDs were constructed on cards manufactured by Baker (9:32-33). System Memory Card 1 was constructed on the correctly manufactured card. System Memory Card 2 was constructed on a reversed image board. These boards function identically and are interchangeable. However, care must be used when following circuit wiring at the edge connectors. Refer to Sec 3.3.1 for a complete description of this discrepancy.

The schematic for System Memory Card 1 is shown in Appendix B, Fig B-4. The edge connector pins shown in parenthesis refer to the reversed image board used for System Memory Card 2.

3.7.2 System Memory Card Construction. The system memory cards contain all control logic and memory ICs necessary for operation. The bottom four rows of each card contain control logic. The remaining rows are dynamic RAM ICs. Fig A-4, Appendix A, shows the IC layout for both cards. Fig 3-6 below shows how the data is organized within the system memory card.

Construction of the system memory cards follows the method used to construct the shared memory cards described in Sec 3.3.2. Power requirements for the dynamic RAM call for three different voltages: +5VDC, -5VDC, and +12VDC (14). These voltages are distributed on each card using the four vertical power buses along the card edges. One bus is used for each voltage and one bus for DC ground. Further description of the power distribution is given in the next section on circuit noise.

3.7.3 Dynamic RAM Noise Considerations. Two characteristics associated with dynamic RAMs in any application are their

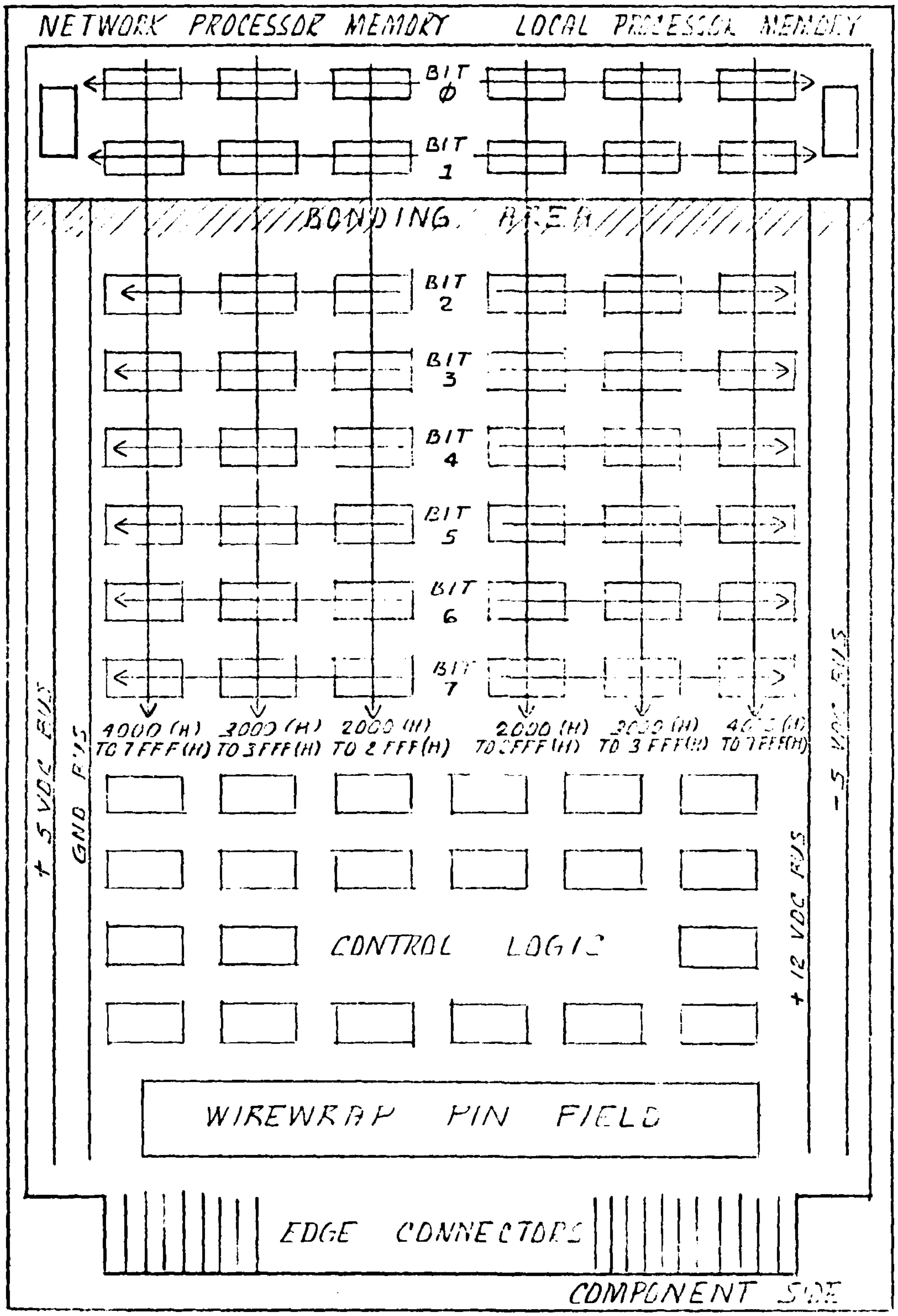


Fig 3-6 System Memory Organization.

need for continual refreshing and their relatively large power requirements (14). Special consideration was given these cards to insure proper noise decoupling and power distribution.

The rapid refresh rate introduces transient signals that can seriously affect the reliability of memory contents. Controlling the effects of these signals was accomplished by designing a decoupling network for the entire card. This design followed the heuristic guidelines established by Intel, Inc, and Texas Instruments, Inc (19:2.1-2.26,20:83-103).

Four typical transient currents are associated with normal dynamic RAM operation. Fig 3-7 displays the current waveforms for these transients. Under normal operation, (A) is the result of internal IC components charging in preparation for a memory cycle. In UNID system memory, this transient occures at the falling edge of the Memory Request (MRQ) signal. Transient (B) is the result of the address buffers turning off after the addresses have been latched onto the memory chip. Point (C) is the level of the steady-state current. Transient (D) results from coupling caused by IC internal capacitance. Finally, (E) shows the charging of internal memory components as the data line prepares for the data. Fig 3-7 also displays typical transient currents associated with the -5VDC current. This is of interest because even though the steady-state current is very low, the peak current can be two orders of magnitude

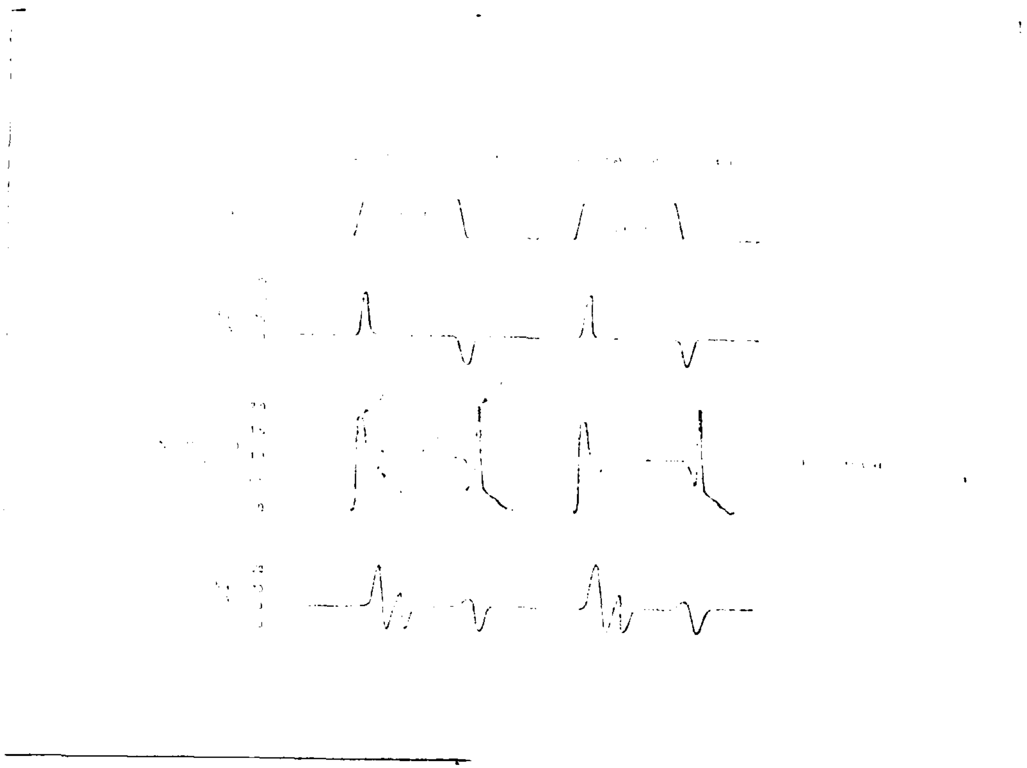


Fig 3-7 Typical Dynamic RAM Transient Currents.

greater.

Transient currents are also associated with +5VDC supplied current. But, these transients are dependent on external loading; the greater the loading, the larger the transient. In this application, the memory IC loading is kept at a very low value by using 8216 bi-directional bus drivers and tri-state IC memories. Thus, transients on this voltage line are within acceptable limits.

Decoupling the transient currents on the +12VDC and -5VDC supplies was accomplished using the decoupling scheme shown in Fig 3-8. Every memory IC has a decoupling capacitor across its +12VDC and -5VDC pins. All capacitors

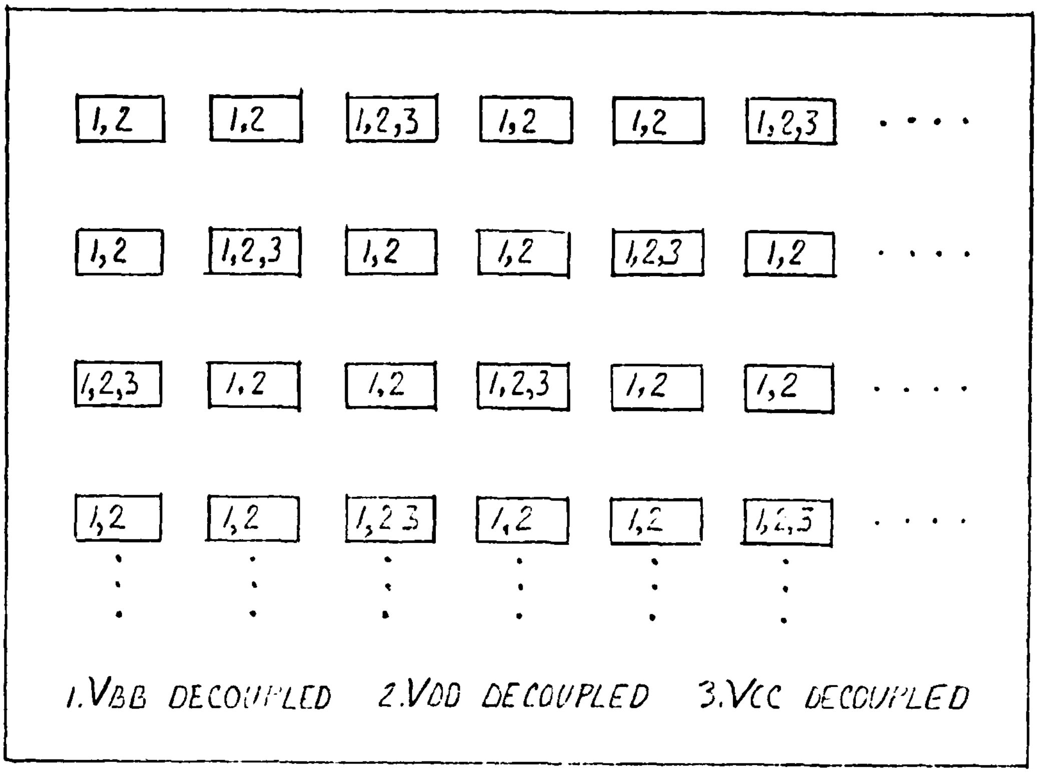


Fig 3-8 System Memory Decoupling Scheme.

are .1 mmf in value. Each control logic IC has its +5VDC pin decoupled also since loading in this section is not as well regulated as in the memory section.

The second problem mentioned above, adequate power distribution, was corrected using a power distribution grid. Power is supplied to each memory IC using the power bus grid shown in Fig 3-9. Here, each power line to a given memory IC is bused both horizontally and vertically. The grid power distribution minimizes the voltage lag that occures when the corresponding current reaches its peak transient value. Excessive voltage lag causes a voltage drop at the pins of the selected memory chips. This

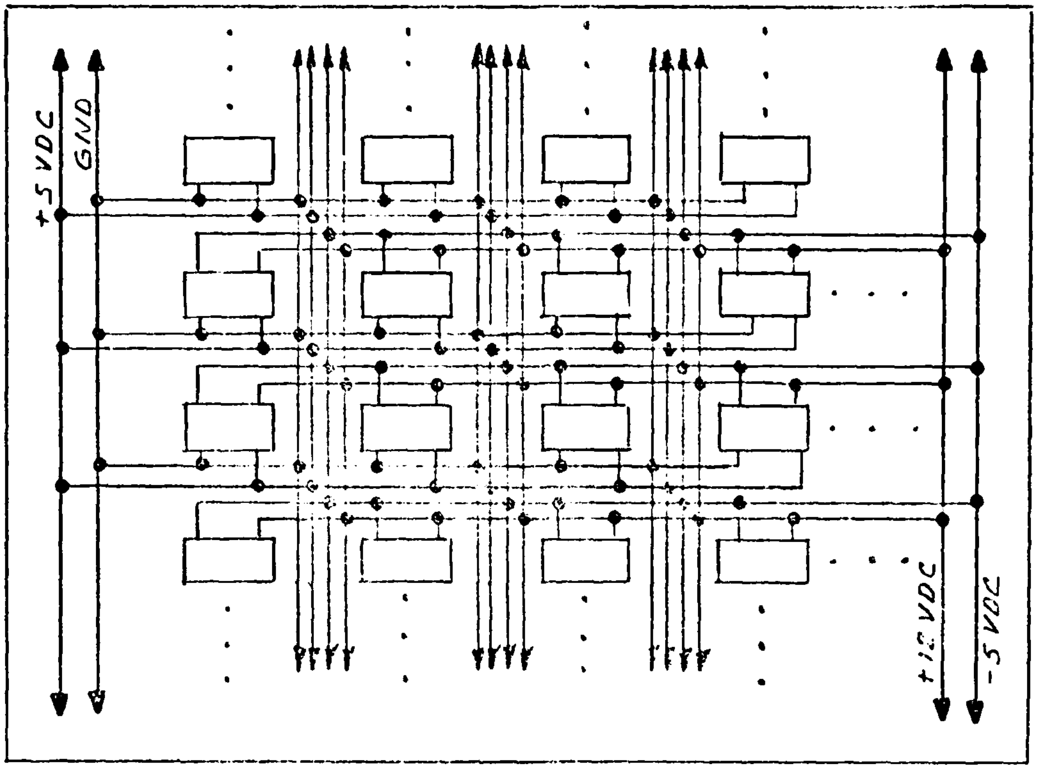


Fig 3-9 System Memory Power Distribution Grid.

short-lived voltage drop makes the chip more susceptable to noise at the critical time when the memory chip is either reading or writing data. A more detailed development of these two noise problems is given later. Refer to Sec 4.4.6.3.

3.8 Network Card Construction.

Network Card 1 was constructed by Brown during his investigation. Network Card 2 was constructed during this investigation. Both cards were built on commercial Z80 card blanks. The cards function identically and are interchangeable. The following sections review card construction, describe the RS-422/423 port configuration, and discuss use of the fiber optic link.

3.8.1 Network Card Completion. Each network card layout is identical to the other. Fig A-5, Appendix A, shows the IC layout. The schematic for these cards is shown in Fig B-5, Appendix B.

3.8.2 Network Card Communication Port. The heart of the network card is the Z80 Serial Input/Output IC. Its flexibility allows high speed serial data transfers, full duplex, under virtually any communications protocol. The configuration used in this investigation was synchronous data transmission at 56 Kbps under the RS-422/423 standard. Channel A was initialized using a software initialization routine. Channel B was not used. The initialization routine is documented in Geist's investigation (11). Futher information concerning the SIO itself is found in Zilog literature (23).

3.8.3 The Network Fiber Optic Link. A fiber optic communications link was included by Hobart for use as the link transmission medium between two UNIDs (10:29). The link was installed between the two network ports on UNID 1 and UNID 2. The hardwired signalling of the fiber optic link is RS-232C. Operating the link under RS-422/423 signalling required slight modification. An interface circuit was designed for each end of the link. Fig 3-10 shows the circuitry used. No modifications were made to the actual

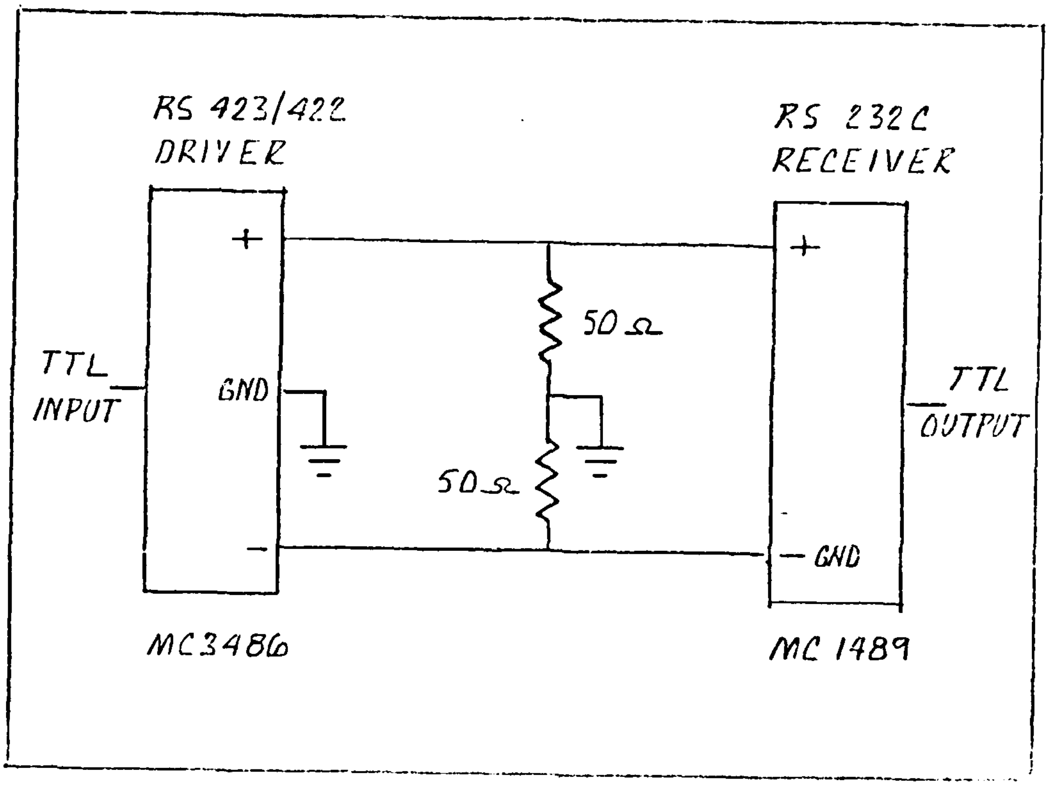


Fig 3-10 Network Port Interface Modification.

hardware of the fiber optic link. Operation and maintenance of the fiber optic link is covered in reference (24).

3.9 DELNET Construction.

Once the two UNIDs were completed and their I/O ports configured, it was a simple matter to construct the prototype DELNET. Building the DELNET required connecting the UNID network I/O ports to each other. The transmission medium used was the modified Valtec fiber optic link described above.

Two equipment configurations were used. The first used ATM-3 terminals as subscribers, as shown in Fig 3-11. Since all local I/O ports were RS-232C compatible, this configuration was assembled by simply plugging in the terminals. The second configuration connected two computers along with the terminals, as shown in Fig 3-12. Again, this was a simple plug-in operation. Notice that Local Channel 1 on each UNID is reserved for loading software from the MCZ-1/25.

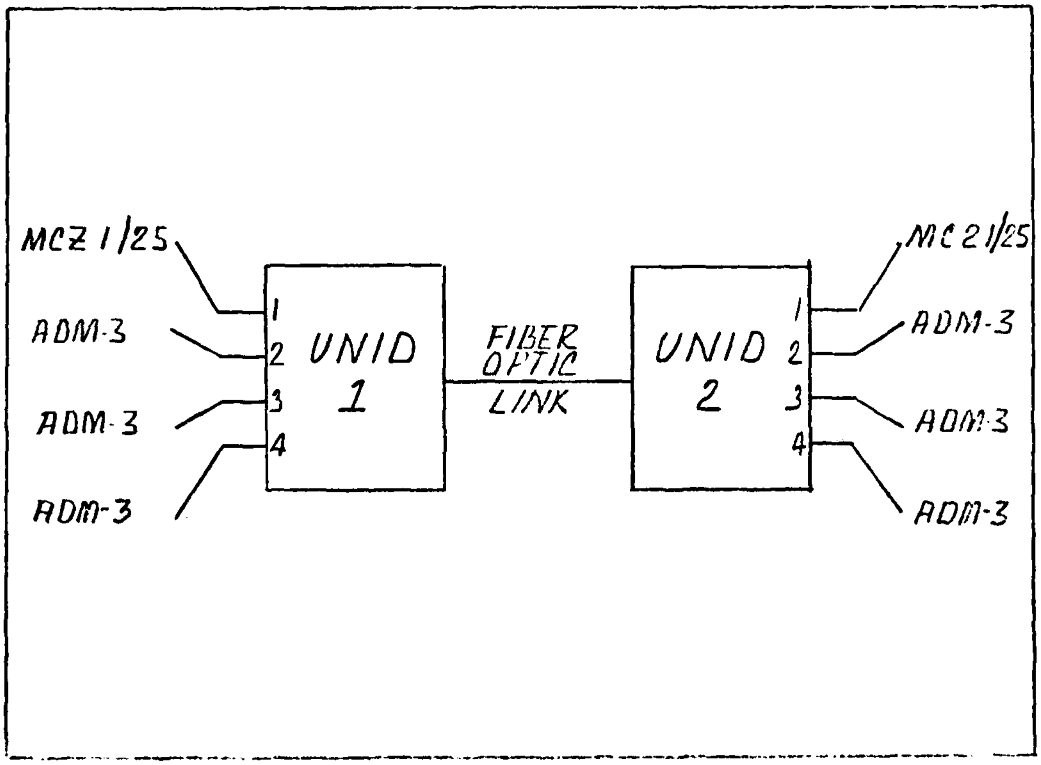


Fig 3-11 Prototype DELNET Terminal Configuration.

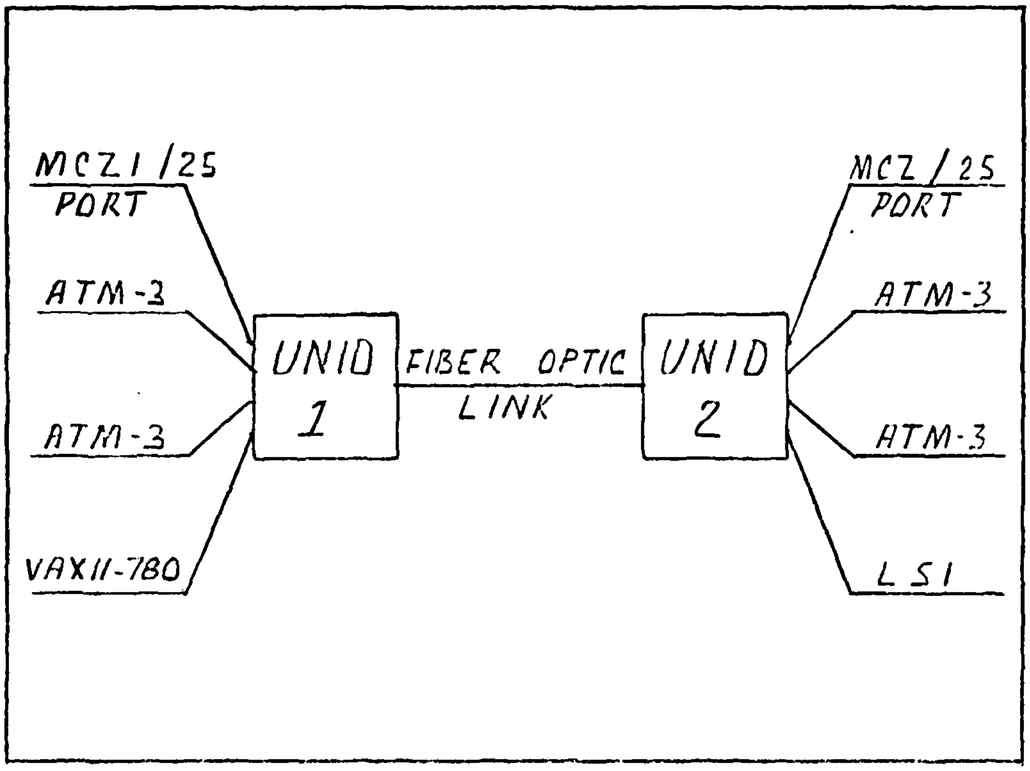


Fig 3-12 Prototype DELNET Computer Configuration.

3.10 Chapter Summary.

This chapter provided a detailed description of the completion of UNID 1 and the construction of the duplicate UNID 2. Each major UNID component was discussed with special attention given to design changes and hardware difficulties. The chapter also described power supply requirements and communication port configurations. The installation and modification of the fiber optic link used as the network transmission medium was also presented. Finally, DELNET construction using the modified fiber optic link was described.

Chapter 4. UNID and DELNET Testing.

The final portion of this investigation involved hardware testing the UNIDs and the prototype DELNET. UNID testing involved verification of wiring on the motherboards and all cards, power distribution tests, functional logic tests, functional memory tests, local and network card tests, and communications channel tests. DELNET testing involved operation of all network components as a whole with several network configurations. The entire testing structure was designed from the bottom-up. Hardware problems encountered are discussed along with their corrective actions.

4.1 UNID Testing Methods.

The tests used to verify UNID operation were structured primarily as incremental bottom up tests. This method was chosen for its several advantages in identifying errors. Once the initial components are checked, any new problem encountered is often caused by either an error in the newly added component or an error in the interaction with the new component. Also, the modularity of the UNID makes this incremental approach particularly attractive. Finally, most of the testing can be completed without developing involved hardware or software support.

The UNID circuit cards were used as the basic incremental component. Since most circuits were

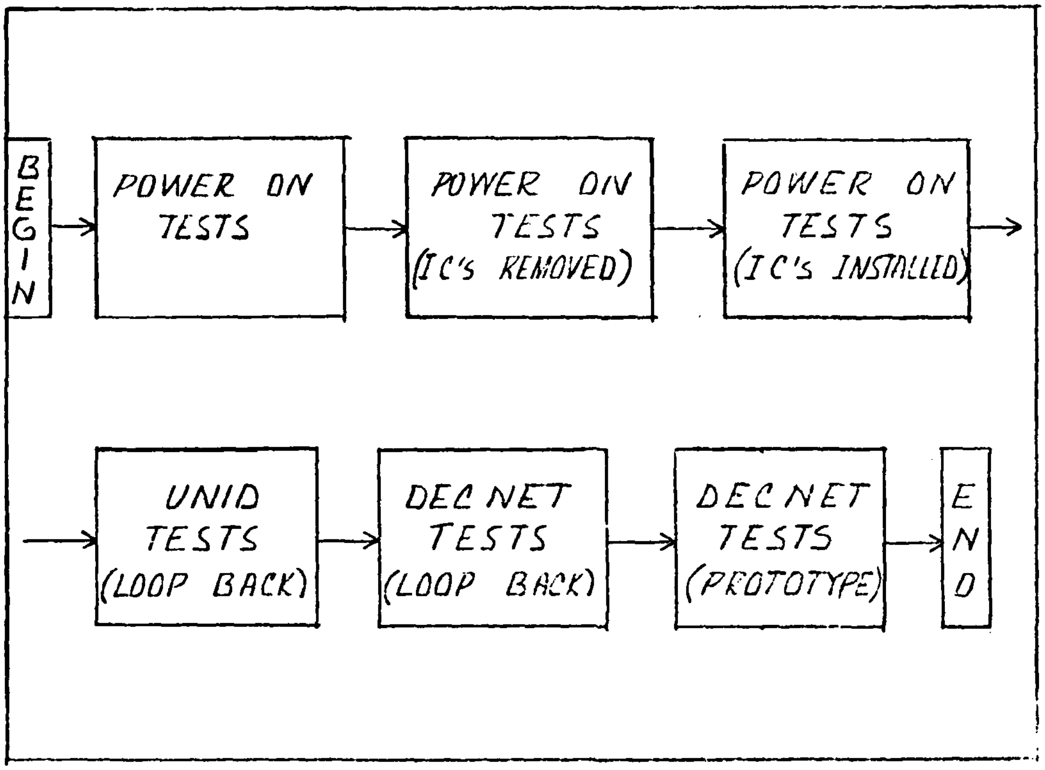

BEGIN
POWER ON TESTS
POWER ON TESTS (IC's REMOVED)
POWER ON TESTS (IC's INSTALLED)
UNID TESTS (LOOP BACK)
DEC NET TESTS (LOOP BACK)
DEC NET TESTS (PROTOTYPE)
END

Fig 4-1 Testing Scheme Flowchart.

functionally tested by Brown and Baker, this investigation limited most testing to checking circuit wiring and verifying card operation. A sequence of tests was established based on these limits. All cards were first given a power-off check as they were completed. The next tests were power-on checks, each card being added and tested before the next card was added. Finally, with both UNIDs completed and checked, the DELNET was formed. Then the entire network was tested. A diagram of this testing scheme is shown in Fig 4-1. The following sections describe each major area of testing and discuss the results.

4.2 Power-off Testing.

Each board used in the two UNIDs was checked for correct wiring before power was applied and before ICs were installed. This check occured after each board was completed. Although these point-to-point wiring checks did not detect all wiring errors, they did insure power was not applied directly to ground, and verified that signals were at least routed to the correct pins. Other wiring problems, such as power appearing on signal lines and signals appearing at other pins in addition to their intended destinations, were less sever faults that could more easily be detected once the chips were installed and power applied. These power-off tests also helped verify the accuracy of the many schematics used in this project.

Very simple methods were used to verify wiring. In some cases the wiring was not very dense and a visual check of each wire was adequate. In other cases the density was too great for visual inspection. In these instances, an audio oscillator and a small speaker were connected in series, as shown in Fig 4-2. When the two circuit points under test had continuity, an audible tone was heard. This method permitted rapid and efficient testing of a large number of circuit points at one time. One test lead was attached to one test point. The other test lead was brushed across the other test points until continuity was found. The pin was identified and checked against the schematic. A large number of wiring errors were detected using this method.

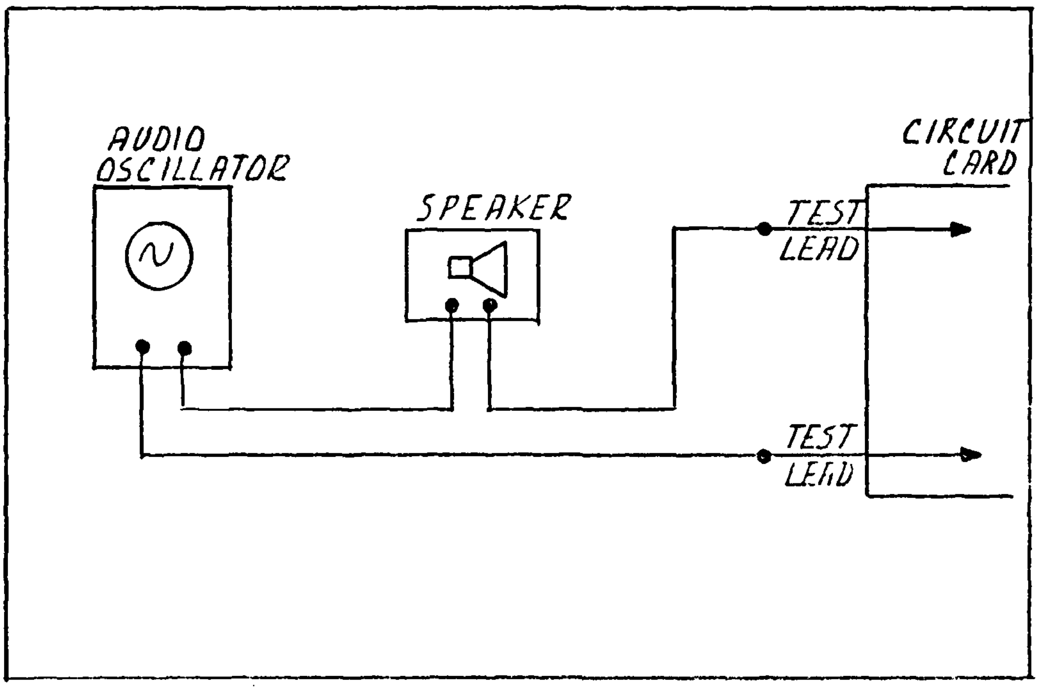


Fig 4-2 Continuity Test Circuit.

4.3 Power-on Testing.

The next series of tests performed were the power-on tests. Power was applied to the completed card without the chips installed. A voltmeter was then used to verify that the correct power appeared at the correct pins. An ammeter was also installed in series with each power supply. If the current needle deflected up scale rapidly this was a sure sign that power was shorted to ground somewhere on the card.

The ICs were then installed and the cards placed in the UNID, adding only one card at a time and checking it thoroughly before adding the next. When power was applied, each chip was checked for excessive heating. This simple test detected numerous wiring errors, shorted chip outputs, and chips that had internal shorts. A number of the shorted chip outputs were due to broken strands of wire that had fallen to the base of the pins during the wirewrapping process.

4.4 Functional Testing.

Once the majority of wiring errors and faulty ICs were detected, each circuit card was tested to insure functional operation. The following sections describe this functional testing for each circuit card. The sequence of the following sections is the order in which the cards were tested.

4.4.1 Local Processor Card Testing. The local processor cards were functionally tested by installing them in the card cage, connecting the reset circuitry, and attaching an ATM-3 terminal as the local processor monitor terminal. These cards contained Baker's EPROM monitor and 4K bytes of dynamic RAM as described in Chapter 3. The functional testing involved resetting the processor card and executing the various commands available in the monitor. (See Appendix E for a description of the commands.) Proper operation of these commands was considered sufficient evidence that the processor was operating correctly. More strigent testing was carried on automatically as cards were added to the UNID and the processor tasks became increasingly complex. For example, using the LOAD command with the local card installed not only tested the on-card memory, but also initialized the USARTs, established the priority interrupts, set the data speed, etc. Each of these functions, directed by the local processor, must function correctly before a file can be loaded. A successful completion of the LOAD command was, therefore, taken as an indication that the local processor functioned correctly.

4.4.2 Shared Memory Card Testing. The next cards added were the shared memory cards. Each local processor monitor command was used again, this time executing the commands within the boundaries of the memory on the shared memory cards. The shared memory was filled entirely with 00(H); then, the entire memory was displayed. This fill-display routine was repeated with different hexidecimal characters until all memory data bits were switched from zero to one to zero. Next, a block of data was moved several times from one location to another within shared memory, then displayed. Finally, a single block of data within shared memory was displayed repeatedly. These three simple tests were used to verify the basic operation and data integrity of shared memory.

The fill-display routine revealed several shared memory problems. Two data-bus lines had faulty wirewraps indicated by a pattern of data bits remaining unchanged after each sequence. A similar symptom revealed that one of the address-line buffers had an open gate.

4.4.3 Network Processor Card Testing. After the local processor and system memory cards past their tests, the network processor card was added. The terminal monitor tests described in Sec 4.4.1 were repeated for the network processor card. Then the fill-display routines in Sec 4.4.2 were performed on the network processor cards alone. Finally, the three cards were tested together. This involved filling a portion of the local processor's on-card RAM, moving that block into the shared memory card, then moving that same block into the network processor's on-card RAM, using the network processor monitor. The entire test was then reversed, beginning at the network processor and ending at the local processor via the shared memory. The sequence was repeated using enough data patterns to test different memory locations. This series of tests revealed a number of wiring errors and faulty ICs within the control logic at the bottom of the shared memory board.

4.4.4 Local Card Testing. The local card was added to the UNID when the above series of tests were acceptable. A file was loaded into the UNID through Local Channel 1 using the MCZ 1/25, the monitor's load command, and existing software. The file's contents and location were verified visually.

Next, Geist wrote a demonstration operating system for the local side of the UNID (11). This program permitted an ATM-3 terminal on one local channel to communicate with any one of the other three terminals. In this test, the demonstration software was loaded through Local Channel 1, stored in local processor RAM, and executed. Channel 1 was then reconfigured as DTE and an ATM-3 connected. With all four terminals connected and the operating system executing in a loop, each terminal sent a string of characters to itself first, then to the other terminals in turn. This test was repeated using a number of different bit rates , including 4800 bps, 9600 bps and 19200 bps. Through this series of tests two faulty USARTs and several jumper plug wiring errors were discovered. After the faults were corrected all four local channels in each UNID past the test.

4.4.5 Fiber Optic Link Tests. At this point the two UNIDs were capable of supporting further development of the local operating system. UNID 1 was relocated for easy access to the VAX and LSI computers. UNID 2 remained at the workbench while the remaining cards were completed. UNID 1, however, required the MCZ 1/25 Software Development System to load software. Moving the MCZ 1/25 was not practical. As a result the fiber optic link was temporarily installed between Channel 1 of UNID 1 and the serial port on the MCZ 1/25. This exercise provided a great deal of practical knowledge in the operation of the fiber optic link. Fig 4-3 shows the configuration used to transfer files to UNID 1.

The fiber optic link came hardwired into the RS-232C DTE configuration (24). UNID 1, Channel 1 was likewise configured by installing a jumper plug in J58 on Local Card

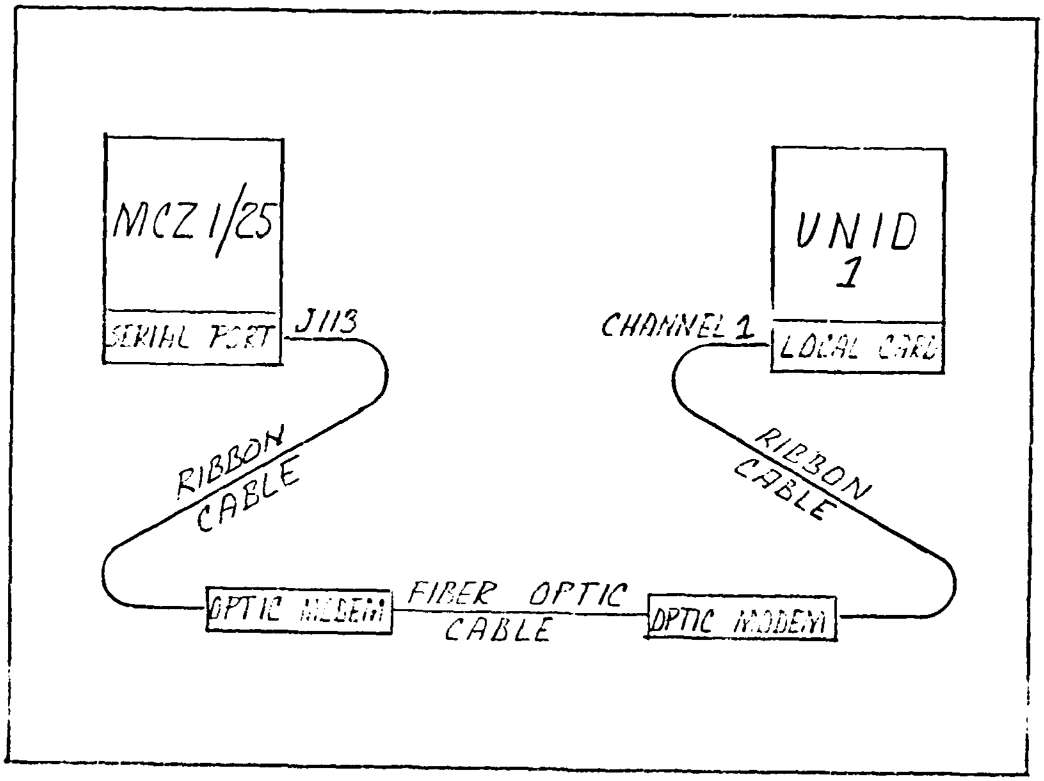


Fig 4-3 File Transfer Configuration.

1. All subsequent file transfers between UNID 1 and the MCZ 1/25 were transfered through the fiber optic link. This channel configuration was used until the fiber optic link was installed as the network communications channel as discussed in Sec 4.4.7.

The following scheme was used to verify the operation of the fiber optic link. A given demonstration program was transfered to UNID 1 via the optic link. The program was then executed. If the program failed to execute as expected, three possible faults were suspected: fiber optic link failure, UNID 1 hardware failure, or software failure. To determine which, the same program was loaded into UNID 2 via ribbon cable, eliminating the fiber optic link. If the program still failed to execute, a hardware or software failure was suspected and action was taken accordingly. If, on the other hand, the program executed correctly then the link was suspected. The fault would most likely involve loss or alteration of data while transferring. Using this test as a performance measure, the fiber optic link performed without a failure.

4.4.6 System Memory Testing. The system memory cards would not operate properly when tested after construction. Basically, four different types of hardware problems were found on these cards: faulty wirewraps and IC sockets, faulty ICs, circuit noise, and timing faults. Each type of problem and the general solutions used will be discussed in the following sections.

4.4.6.1 Wirewrap and IC Socket Faults. Of all the wirewrap circuit cards constructed, the shared memory and system memory cards are the most densely wired. Of these two, the system memory cards are the most dense. This implies that these cards have a very large number of hand applied wirewraps. Indeed, the system memory card has approximately 4000 individual wraps. With such a large number of wirewraps on a single board, a few faulty connections can cause great difficulty. Detection of numerous such faults can become a laborious chore.

The faulty wirewrap discussed here is one where the wire is wrapped around the post using insufficient tension on the wire. This fault is usually associated with wirewraps applied using a manual wirewrap tool. Most faulty wraps will operate satisfactorily for a period of time. But, as the circuit ages these wraps have a tendency to develop corrosion between the wire and the post. If corrosion forms, the wrap will develop a high resistance joint eventually opening completly.

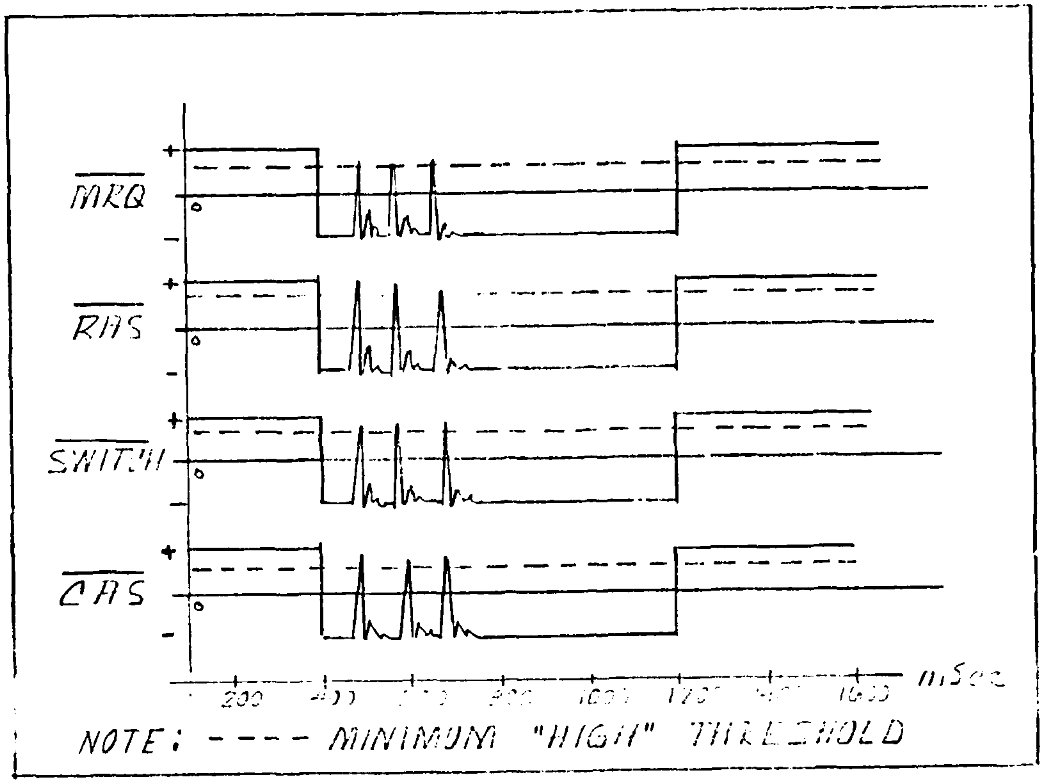
A high resistance joint is very difficult to detect since it requires precise ohmmeter measurements using less-than-perfect surface contact test leads. An open is somewhat easier to detect. A signal measurement is made on one side of the card. The measurement is then repeated at the same point on the opposite side of the card. If the two measurements are different, then the wirewrap or socket is suspected. Three common faults displayed this symptom: corrosion on the faulty wrap, corrosion on the IC pins, and poor contact between the IC pins and socket. Once detected, correcting these faults is simple. The faulty wraps were replaced. The faulty socket or IC was either replaced, or cleaned and adjusted. Unfortunately, there is no way of guaranteeing that these problems will not recur. Being aware they might exist is important for future troubleshooting.

4.4.6.2 Integrated Circuit Faults. An IC fault occurs if a chip fails to perform according to its truth table or specified function. Typically, an IC fault where the output remains high under all conditions is corrected easily. The chip is removed and replaced. A fault where the output remains low under all conditions is more difficult to diagnose. A typical low fault may be caused by an internal short, an external short, or an internal short in the next stage. Detecting the latter fault can be particularly difficult in complex circuits such as memories.

One effective way to isolate the fault involves removing and testing individual chips. A suspected IC is removed and installed on a design test board such as the Elite 1 manufactured by E and L Instruments, Inc. A simple circuit is built and the IC's truth table is observed. If the truth table is correct, another chip is removed from the board and tested. If all associated chips test satisfactorily, the fault is typically with the wiring.

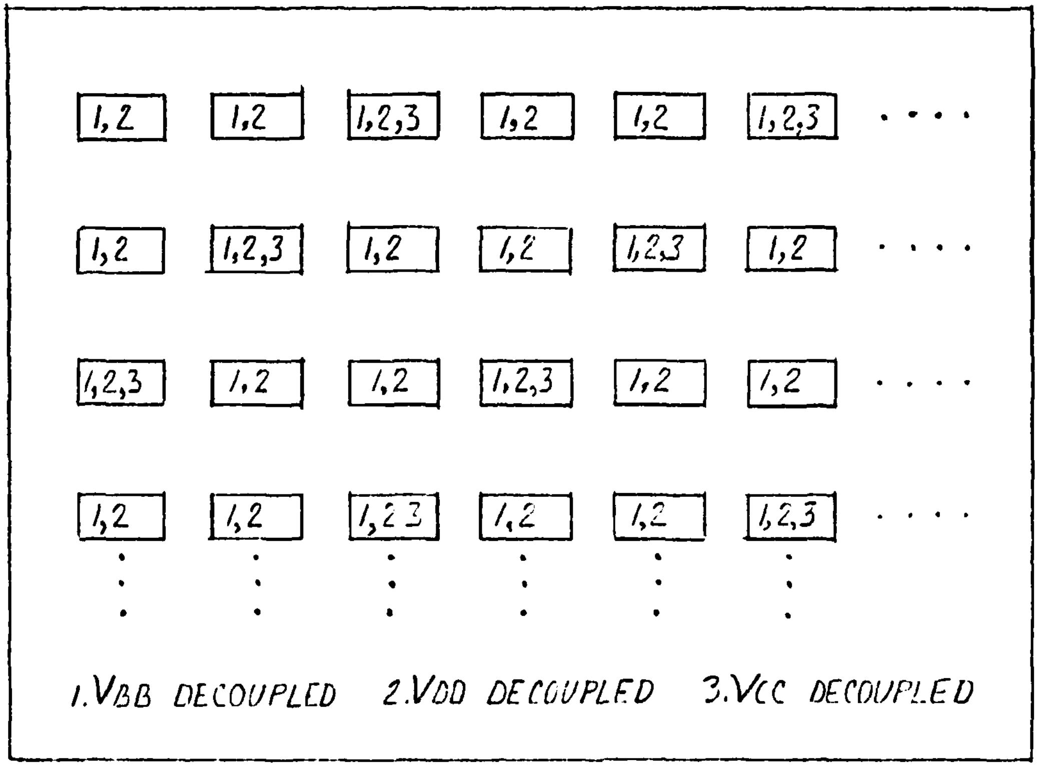
4.4.6.3 Circuit Noise Faults. The circuit noise faults observed on the system memory boards were of two types: transient noise and poor power distribution. These faults caused the memory to be extreemly unreliable. Both faults are easily detected using an oscilloscope. The benifits of using a storage oscilloscope for these types of problems cannot be overemphasized. The oscilloscope captured specific events and stored them for photographing and comparing. Further information about a suitable oscilloscope can be found in reference (25).

Fig 4-4 shows a control signal, such as Memory Request (MRQ), without proper decoupling. Also shown are other



NOTE: ---- MINIMUM "HIGH" THRESHOLD

Fig 4-4 Control Signals with Noise.



1,2	1,2	1,2,3	1,2	1,2	1,2,3	....
1,2	1,2,3	1,2	1,2	1,2,3	1,2	....
1,2,3	1,2	1,2	1,2,3	1,2	1,2	....
1,2	1,2	1,2,3	1,2	1,2	1,2,3	....
⋮	⋮	⋮	⋮	⋮	⋮	

1.VBB DECOUPLED 2.VDD DECOUPLED 3.VCC DECOUPLED

Fig 4-5 System Memory Decoupling Matrix.

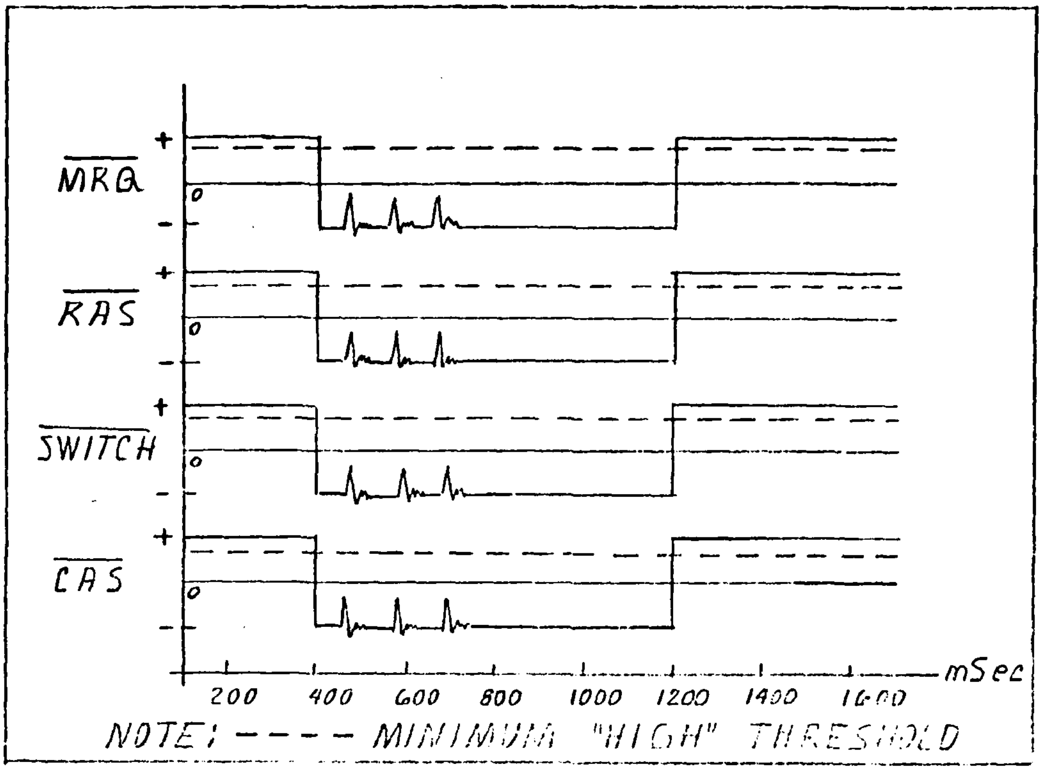


Fig 4-6 Control Signals after Decoupling.

control signals associated with Memory Request. Notice that the pattern of noise components is carried throughout the other signals. These noise signals have peak values of about 400 mV. Under these conditions the system memory cards could not store data accurately.

Fig 4-5 shows the decoupling matrix used on the system memory cards, as described in Sec 3.7.3. The same set of signals shown in Fig 4-4 before decoupling is shown in Fig 4-6 after decoupling. Here the noise patterns have been reduced to 100 mV or less, an acceptable level for reliable memory operation (19).

Using the same oscilloscope techniques on the supply

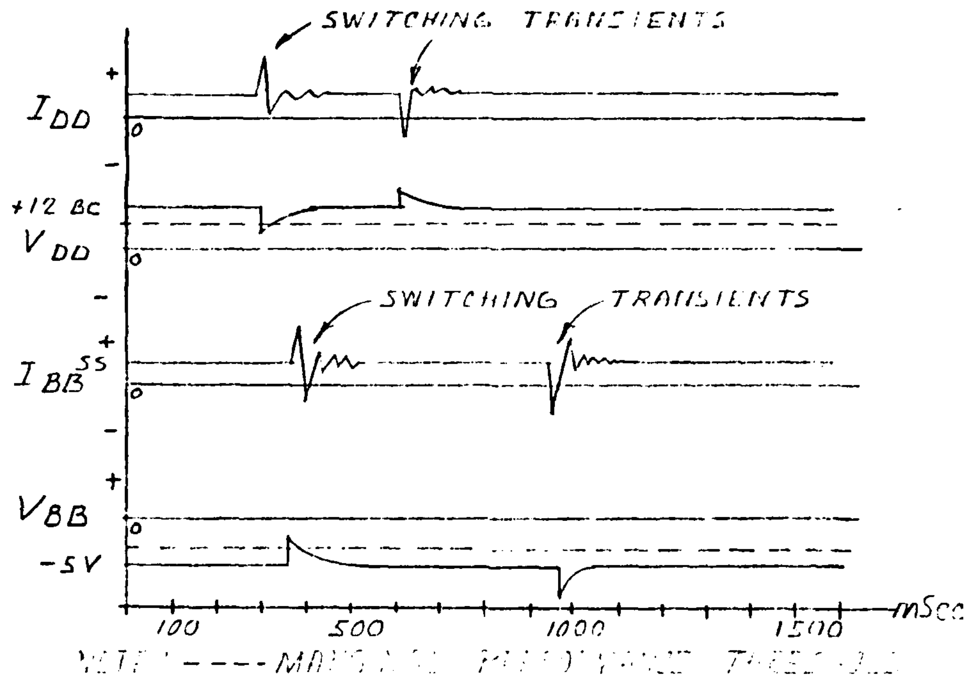


Fig 4-7 Vopltage Lag in Memory.

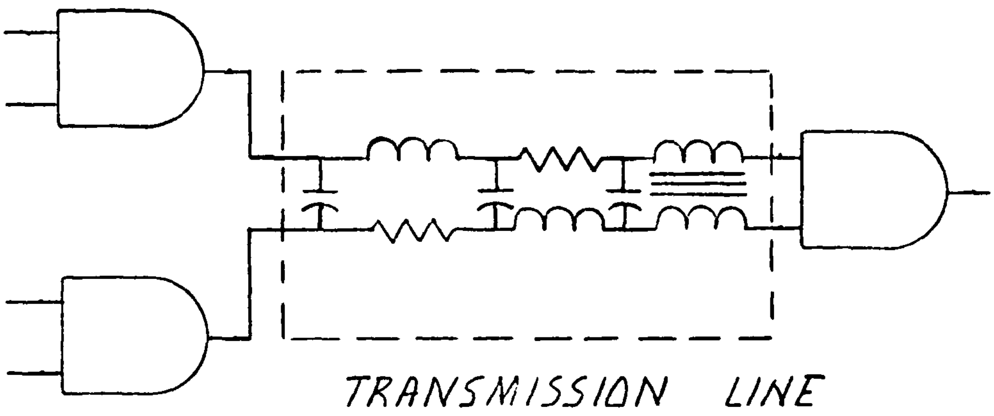
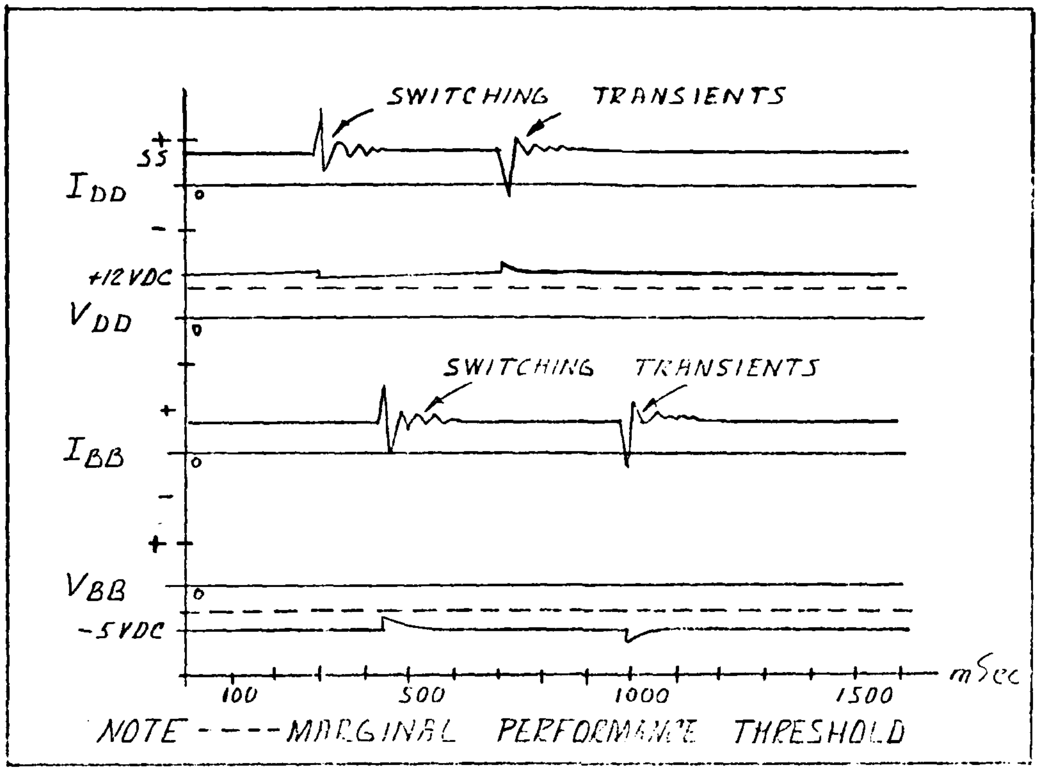


Fig 4-8 Transmission Line Characteristics.



NOTE ----MARGINAL PERFORMANCE THRESHOLD

Fig 4-9 Memory Voltage Lag with Power Grid.

voltages appearing at the memory chip pins reveals the second problem, voltage lag. Fig 4-7 shows the -5VDC and +12VDC power at a memory chip during a memory cycle. Also shown are the typical transient currents for the same memory cycle. Notice how the pin voltages drop when the associated transient current peaks. This is a true voltage lag caused by the transmission line characteristics of the power bus system shown in Fig 4-8.

An effective power bus grid, as shown in Fig 3-9, minimizes the voltage lags (19). Without controlling this lag each memory chip becomes marginally operational during the various memory cycles. Typically, the voltage appearing at the IC will drop making the IC more susceptable to noise while it is reading or writing data. After the power bus grid was installed, the voltages were again monitored. Fig 4-9 shows these voltages. Notice that the voltage drop is less during the memory cycle.

4.4.6.4 Memory Timing Faults. After the above faults were detected and corrected, the system memory cards still failed to be completely reliable. During fill and display memory operations, a set pattern of unchanging data usually occured. Although this set pattern occurred at only 21 memory locations out of 32,000, the pattern had to be removed before the memory could be used. At this point the only operating feature of system memory not investigated was the memory cycle timing. A test was devised using the Model 464 storage oscilloscope and the HP Model 1600A logic state analyzer to measure timing signals.

Fig 4-10 displays the correct timing diagram for the system memory cards. These waveforms are measured with reference to the leading edge of the selected memory address and to the falling edge of the T3 clock pulse. The times shown in this figure represent the typical delays for proper memory operation.

The timing signals were measured using the circuit shown in Fig 4-11. The logic analyzer was connected to the 16 address lines at the input to the 74LS367 buffers (26).

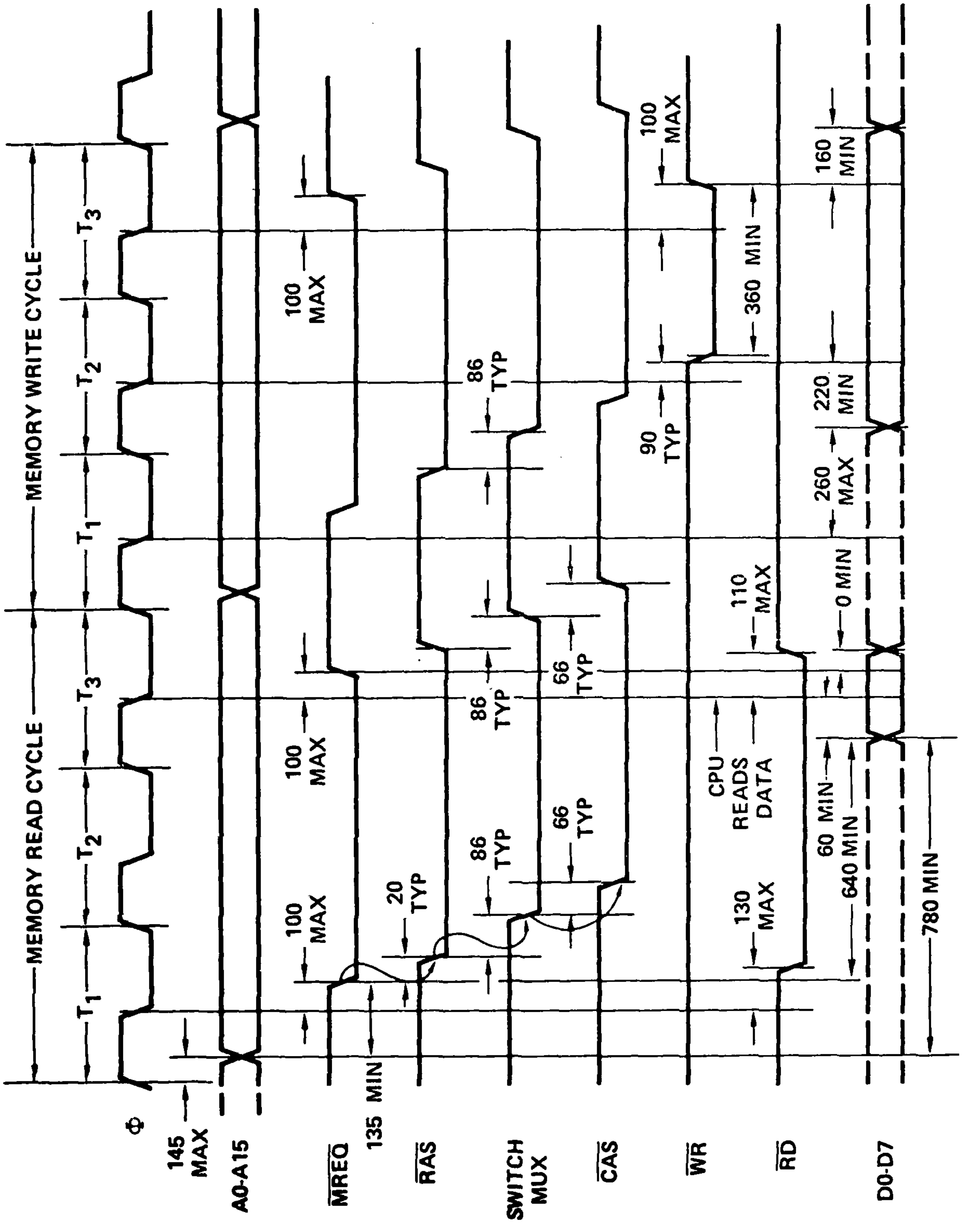


Fig 4-10 Correct System Memory Timing Signals.

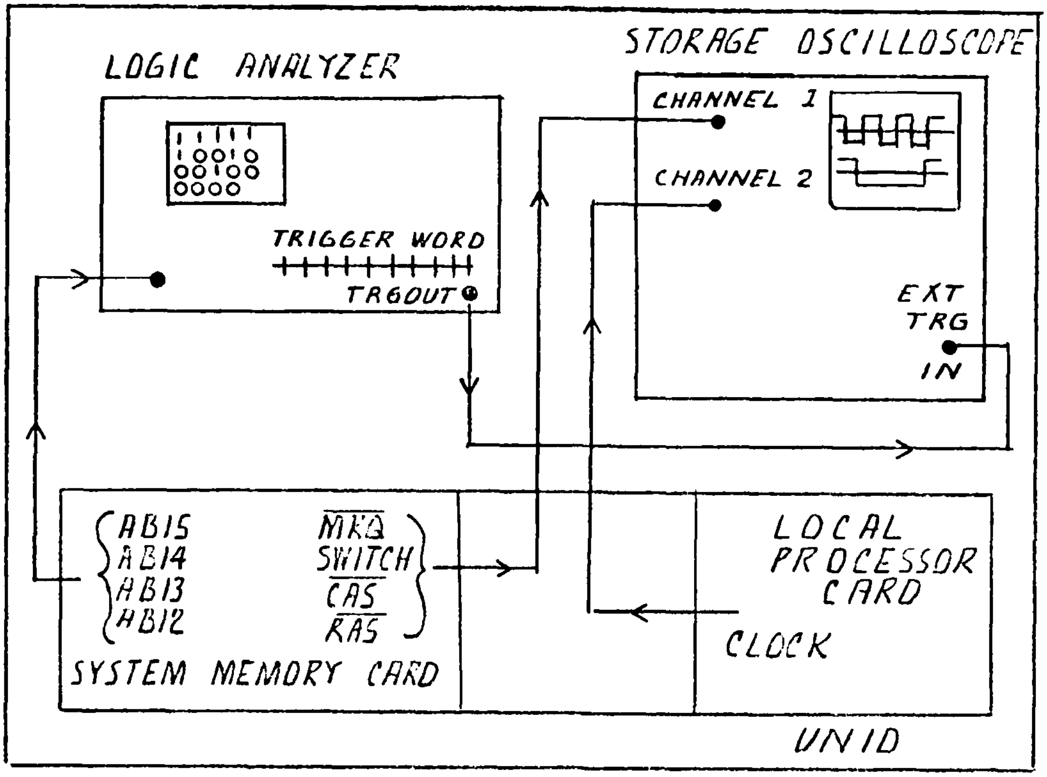


Fig 4-11 System Memory Test Configuration.

The clock signal was taken from A38-2 on the local processor card. No qualifiers were used for this test. The logic analyzer was operated in the Trigger Word mode. The selected trigger word was a single system memory address. When the selected add:ess was accessed by the local processor, the logic analyzer generated a Pattern Trigger Pulse which was fed into the storage oscilloscope as an external trigger. The pattern trigger fired the oscilloscope at the correct time to catch the desired timing signals. The Pattern Trigger established the leading edge address reference. Displaying the clock puises on one of the two available oscilloscope channels established the

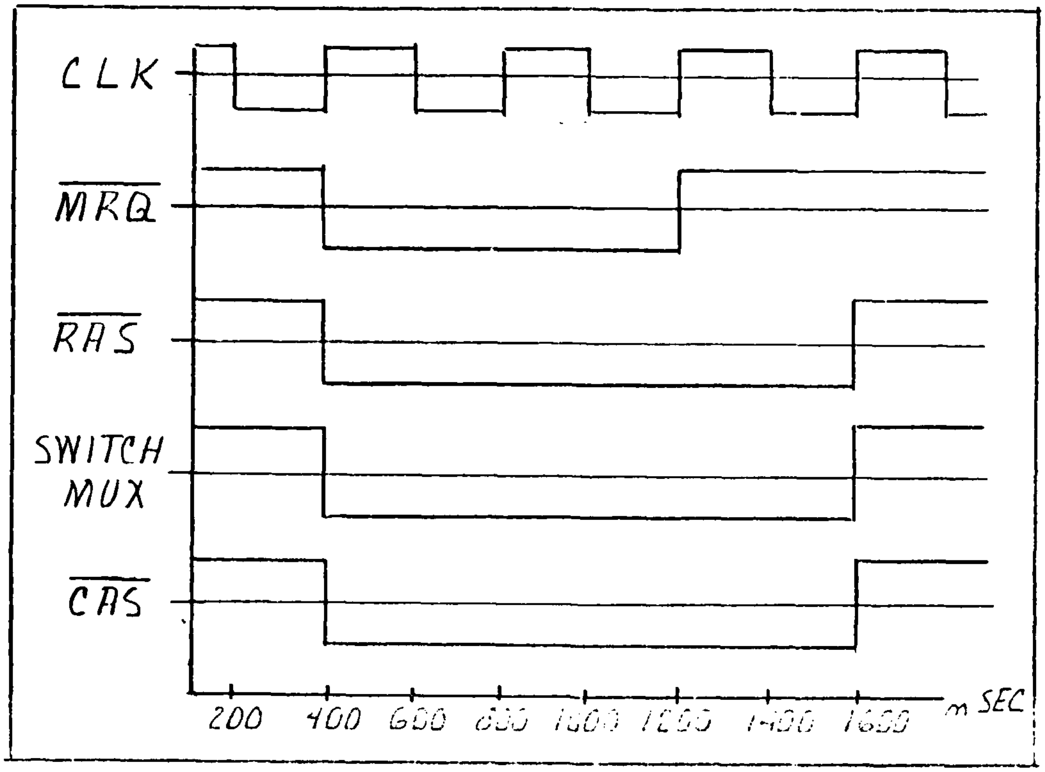


Fig 4-12 Incorrect System Memory Timing Signals.

falling edge reference of T3. The remaining oscilloscope channel was used to display each of the desired timing signals, using the Multiple Display Storage function. Fig 4-12 displays the timing signals measured on the system memory card.

This test revealed the existence of a timing fault. Comparing Fig 4-10 with Fig 4-12, notice that the RAS, SWITCH, and CAS signals are about 1200 nS in duration as opposed to the required 800 nS. These longer control signals during the memory write cycle were overlapping the M1 memory cycle. When this happened the data in shared memory was being overwritten by the data ment for the M1

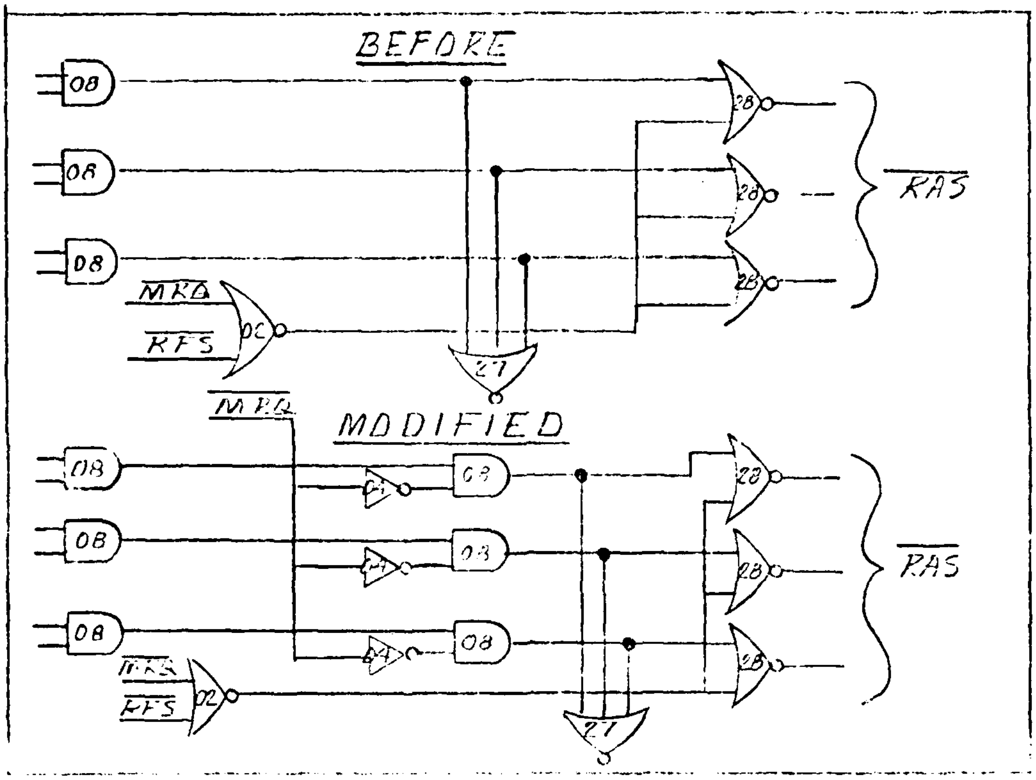


Fig 4-13 System Memory Circuit Modification.

cycle. It appears that the consistency of the data pattern error in shared memory was the result of an M1 instruction word being written into shared memory at exactly the same time in each M1 cycle.

This problem was solved by limiting the duration of the RAS signal at J18 and J20 on Fig B-4, Appendix B. Fig 4-13 shows the circuits before and after modification. An inverted MRQ was ANDed with the RAS. This limited RAS duration to that of MRQ, 800 nS. Limiting RAS in this manner subsequently limited SWITCH and CAS. Once these signals had the correct duration, the system memory cards operated correctly. The system memory cards were then

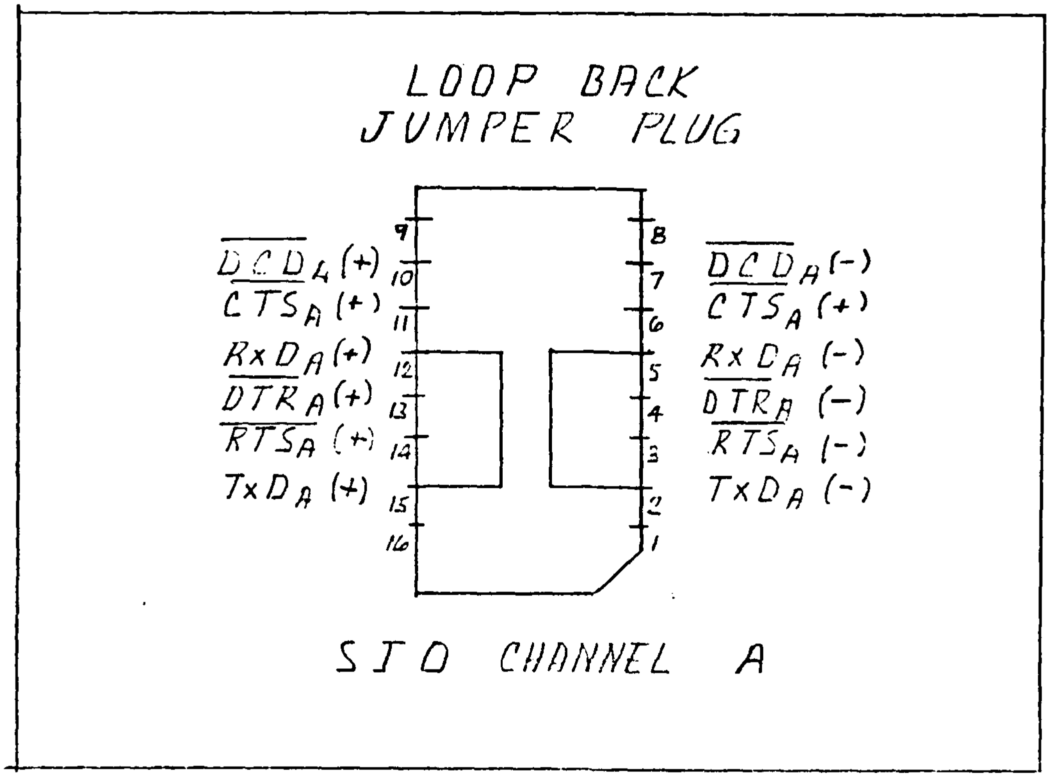


Fig 4-14 Network Loopback Jumper Plug.

tested in the same manner as the shared memory cards. See Sec 4.4.2 for a description of this functional testing.

4.4.7 Network Card Testing. The network cards were tested in the UNID using previously developed software (9:59-64). The network card was configured to loop back the output into the input at the RS-422/423 line drivers and receivers by simply replacing the jumper plugs. Fig 4-14 shows the loop back jumper plug used. The test was run with a message inputted at one local channel and outputted at another local channel. The message actually went from the local card, through the local and network processors, into the network card, looped back, and returned to the local card at a different local channel. In this manner, the entire UNID was tested each time a message was sent. Each UNID tested without a failure.

4.5 DELNET Test.

A test similar to the network card tests was used to test the DELNET. The jumper plugs for RS-422/423 operation were installed and the network operating system was modified slightly, changing the message routing addresses (11). Next, the two UNIDs were connected together using the fiber optic link as the network link. The test was run again with a message inputted at a channel on Local Card 1 in UNID 1 and outputted at a channel on Local Card 2 in UNID 2. The DELNET configuration for this test is shown in Figs 3-11 and 3-12. The DELNET performed in this configuration without a fault. Later, when software was available, this basic test was repeated using the Vax and LSI computers. In this test, the two computers passed simple messages between themselves and to terminals connected to the network. Documentation of these tests and the associated software is contained in Geist's work (11). The DELNET, in this configuration, performed these tests satisfactorily.

This test required the fiber optic link to be reconfigured as RS-422/423. Fig 3-10 shows the circuitry necessary to make this change. No hardware alterations were made on the fiber optic link. All required signal modifications were made by the added circuitry.

4.6 Chapter Summary.

This chapter describes the testing performed on the UNIDs and the prototype DELNET. The testing scheme chosen was organized from the bottom up. The types of tests used were power-off tests, power-on tests, and functional tests. Also discussed in this chapter were test equipment configurations, types of circuit faults encountered, and correction of these faults. Finally, the chapter concludes with an evaluation of the performance of the two UNIDs when they were connected into the prototype DELNET.

Chapter 5. Summary and Recommendations.

The objective of this investigation was implementation of a prototype packet-switched ring local network using dual micorcomputer based message processors for network nodes. This involved completion of two message processors (UNIDs), modification of hardware, construction of the protorype network (DELNET), and testing. The software used in this investigation either existed or was developed by Geist, a co-investigator. This chapter summarizes the results of these efforts and offers recommendations for further study.

5.1 UNID Completion.

When this investigation began, one prototype UNID was partially completed. Enough circuitry was present to test the basic UNID functions. During this investigation, the prototype was completed and a second UNID built. UNID memory was expanded to its 64K byte limit. The shared memory card added a single block of 32K bytes, available to each processor. The system memory card added two blocks of 24K bytes, one block available to each processor. The number of local I/O ports for each UNID was increased from two to four. Thus, each UNID is capable of interfacing four separate subscribers to the network. A second set of local and network monitor EPROMs was generated and installed in the second UNID. Finally, enough cables were manufactured to connect eight subscribers into the network at the same time.

5.2 Hardware Modifications.

Completing the UNIDs required several hardware modifications. The most involved modification added a power distribution grid and decoupling network to each of the four memory cards. Each of the memory cards was also extended to hold the required number of IC memory chips. The local card I/O port line receivers were modified to accept a wider range of port signalling routines. Finally, a memory timing fault was detected and corrected on the two system memory cards.

5.3 DELNET Construction.

Construction of the prototype DELNET was accomplished easily, once the UNIDs were functional. The network I/O ports were connected together using a commercial fiber optic communication link. Since the link was hardwired into the RS-232C DCE configuration, a circuit was added converting the link to the required RS-422/423 operation. Terminals and two computers were connected to the DELNET as subscribers using the RS-232C DTE configuration.

5.4 UNID and DELNET Testing.

Each UNID and the DELNET were tested using an incremental bottom up testing routine. Most of the UNID testing was limited to detecting construction errors.

Generally, the construction involved expanding or duplicating existing circuit cards. Functional testing of circuit groups was completed and documented in previous UNID investigations. However, the difficulties encountered on the system memory cards required extensive testing to detect and correct. Special tests were used to detect various noise, power, and timing faults existing in these cards. The UNID, as a whole, was tested using software developed by Geist.

Once the UNIDs were functioning properly, testing the prototype DELNET was the next matter. The tests were similar to the UNID network card tests except each UNID transmitted across the link to the other instead of looping back. These tests demonstrated that terminals and computers can communicate with each other across the prototype DELNET using the simplified interrupt, routing, and packeting schemes developed by Geist.

5.5 Recommendations.

The recommendations involve further hardware development for the UNID and DELNET. Implementation of the Z80A processor is still a requirement of the original design. This involves replacing each Z80 MCB with the faster Z80A board. Since the other components of the UNID can operate at faster speeds than used currently, this would be a straight foward means of increasing UNID throughput while decreasing response time.

The difficulties encountered with dynamic RAM should be eliminated altogether. Circuit coupling and switching noise will always be a problem for the wire wrapped system memory cards. These problems may reappear when the Z80 processor is upgraded to the faster Z80A. Additionally, the problems of corrosion at the wire wrap terminals can become much worse. A new system memory board could be designed using static RAM memories with greater density. Static memory chips of up to 4K X 1 Byte are becoming available. Using these ICs, the entire 48K bytes of system memory could be built with only 12 chips.

Another recommendation deals with the network card. The original design goal for the network data rate is 1.5 Mbps. The Z80A-SIO can function at 880 Kbps in half duplex mode. The desired rate could be reached using the Signetic's 2652 Multi-protocol Communication Controller (MPCC). The MPCC data rate specification is 2 Mbps and contains all Z80A-SIO features. It can also interface with an 8-bit or 16-bit data bus. Thus, this addition would be compatible with 16-bit message processors currently being developed.

Finally, the performance of the UNID and DELNET to this point justifies construction of additional UNIDs and expansion of the DELNET to other computers in the Digital Engineering Laboratory. Although this requires much more software than is currently available, a long lead time is needed to construct just one additional UNID. Sound planning, spreading construction over several subsequent investigations, will ensure DELNET hardware is available as software is completed.

Bibliography

1. Liebowitz, Burt H. and John H. Carson. Tutorial Distributed Processing COMPCON Fall 77. New York: IEEE Press, 1977.

2. Tanenbaum, Andrew S. Computer Networks. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1981.

3. Kuo, Franklin F. Protocols and Techniques for Data Communication Networks. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1981.

4. Thurber, Kenneth J. Tutorial: Distributed Processor Communication Architecture. New York: IEEE Press, 1979.

5. Metcalfe, Robert M. and David R. Boggs. Ethernet: Distributed Packet Switching for Local Computer Networks. Technical Report CSL-75. Xerox, Inc., Palo Alto, California. May 7, 1975.

6. 1842 EEG/EEIC An Engineering Assessment Toward Economic, Feasible, and Responsive Base-Level Communications through the 1980's. Technical Report TR 78-5. Richard-Gebaur AFB, Missouri. October 31, 1977.

7. Sluzevich, Sam C. Preliminary Design of a Universal Network Interface Device. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1976. (AD-A064-059)

8. Brown, Eric F. Prototype Universal Network Interface Device, MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1979.

9. Baker, Lee R. Prototype and Software Development for Universal Network Interface Device. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, December 1980.

10. Hobart, William C. Design of a Local Computer Network for the Air Force Institute of Technology Digital Engineering Laboratory. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, March 1981.

11. Geist, John W. Protocol Development for the Universal Network Interface Device. MS Thesis. Wright-Patterson AFB, Ohio: School of Engineering, Air Force Institute of Technology, to be published.

12. 9022-78R. An Introduction to SADT, Structured Analysis and Design Technique. Waltham, Massachusetts: SofTech, Inc., November 1976.

13. Schwartz, Mischa Computer-Communication Network Design and Analysis. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1977.

14. Motorola, Inc. MCM4027A Dynamic Random Access Memory. Phoenix, Arizona: Motorola, Inc., 1978.

15. Zilog, Inc. Z80-MCB Hardware Uses's Manual. Cupertino, California: Zilog, Inc., January 1978.

16. Powermate,Inc., ESM-200 Switching Power Supply. Hackensack, New Jersey: Powermate, Inc., 1980.

17. Texas Instruments, Inc. The TTL Data Book (Second Edition). Dallas, Texas: Texas Instruments, Inc., 1976.

18. Intel, Inc. 2114A Static Random Access Memory. Santa Clara, California: Intel, Inc., 1980.

19. Intel, Inc. Memory Design Handbook. Santa Clara, California: Intel, Inc., 1975.

20. Morris, Robert L. and John R. Miller. Designing with TTL Integrated Circuits. New York, New York: McGraw-Hill Book Co., 1971.

21. Zilog, Inc. PROM User's Manual. Cupertino, California: Zilog, Inc., 1979.

22. Texas Instruments, Inc. The Linear and Interface Circuits Data Book. Dallas, Texas: Texas Instruments, Inc., 1973.

23. Zilog, Inc. Product Specification Z80-SIO. Cupertino, California: Zilog, Inc., March 1978.

24. Valtec, Inc. Model RSK-D2 RS-232C Fiberoptic Data Link Instruction Manual. West Boylston, Massachusetts: Valtec, Inc., 1980.

25. Tektronix, Inc. 464 Storage Oscilloscope Instruction Manual.
Beaverton, Oregon: Tektronix, Inc., June 1976.

26. Hewlett-Packard, Inc. 1600A Logic State Analyzer. Colorado Springs, Colorado: Hewlett-Packard, Inc., 1978.

Appendix A

Circuit Card Layouts

This appendix contains a collection of diagrams illustrating the IC layout for five circuit cards used in the UNID, including the motherboard. The local and network processor cards are not included here, but are well documented in the Z80 MCB User's Manual (15).

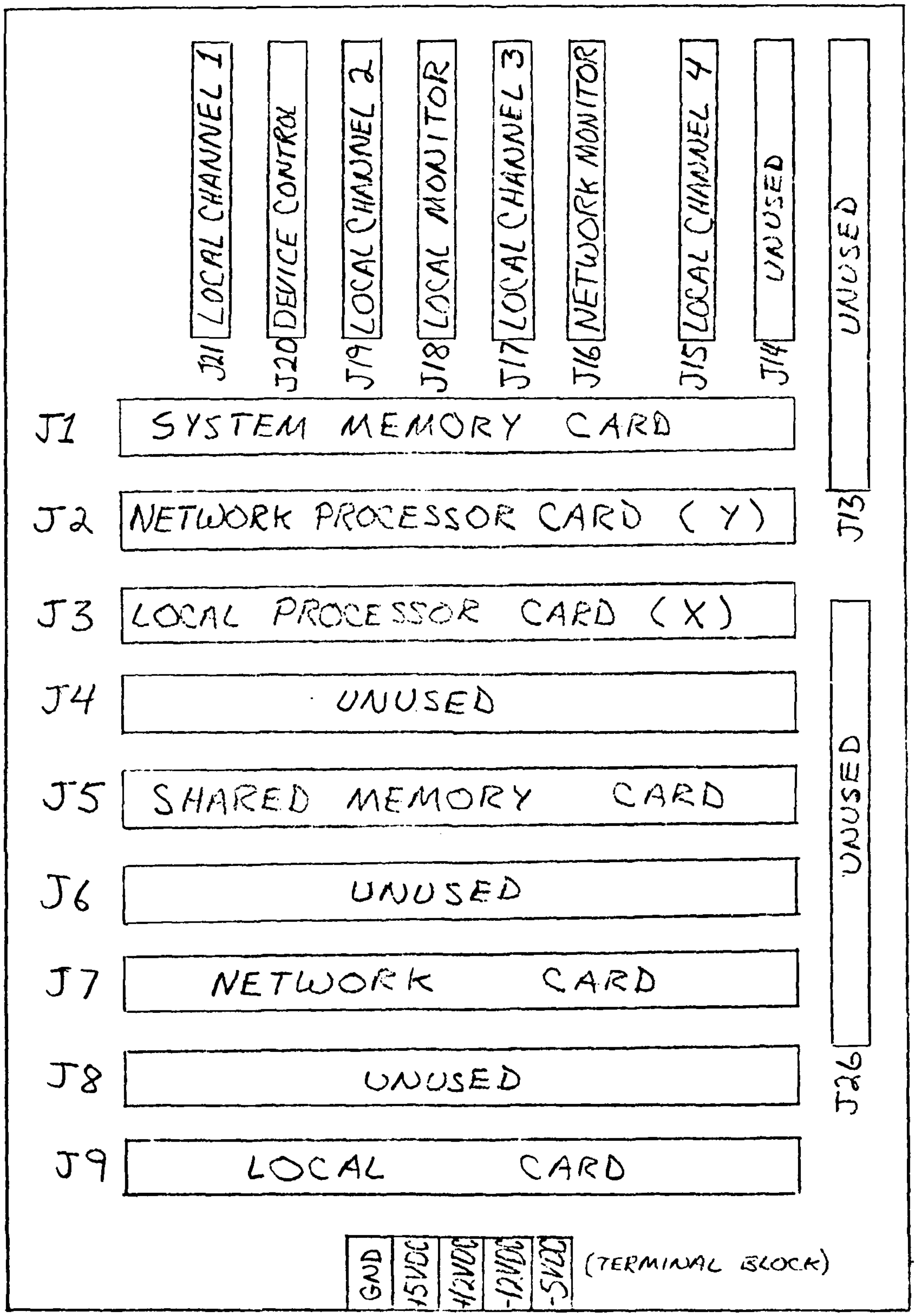
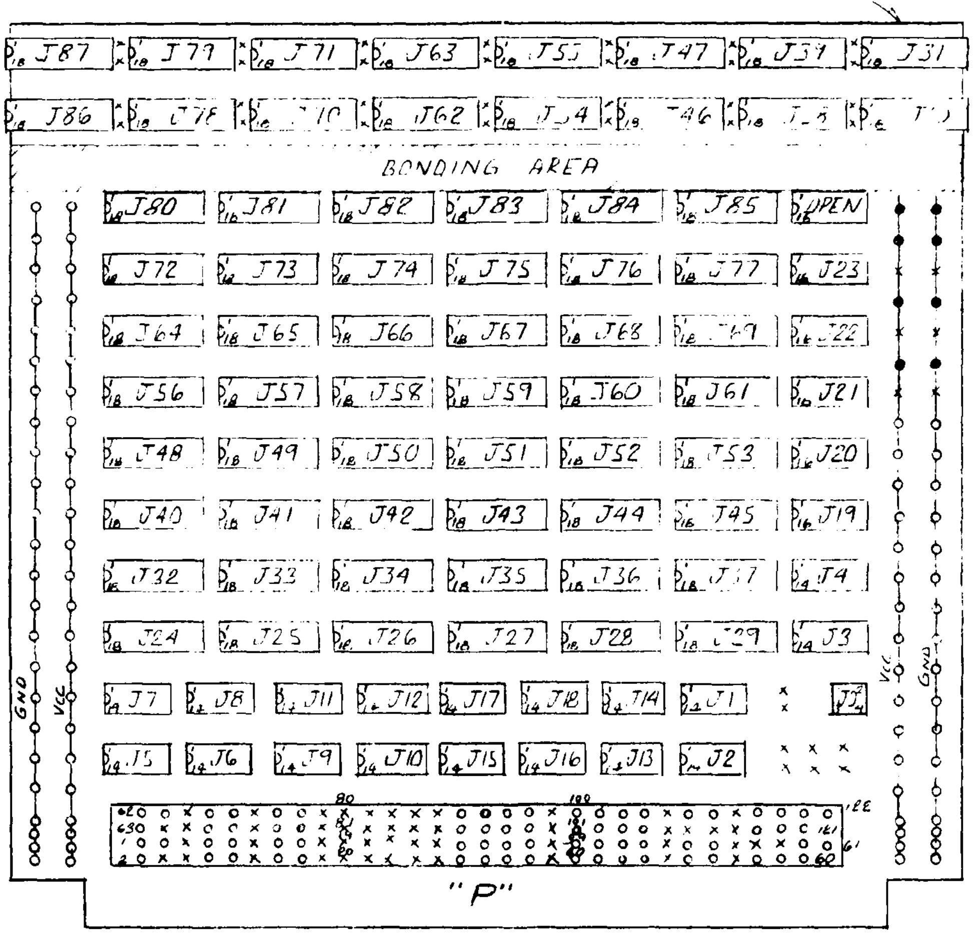


Fig A-1 Motherboard Layout.

BOARD EXTENSION



SHARED MEMORY CARD (JS-UNID #1)
(WIRE WRAP SIDE)

LEGEND

X - NO WW PIN

○ - WW PIN INSTALLED

● - WW PIN TO BE INSTALLED

ALL SOCKETS & CHIPS HAVE KEY FACING LEFT EDGE OF WIREWRAP SIDE OF BOARD UNLESS NOTED OTHERWISE

Fig A-2 Shared Memory Card Layout.

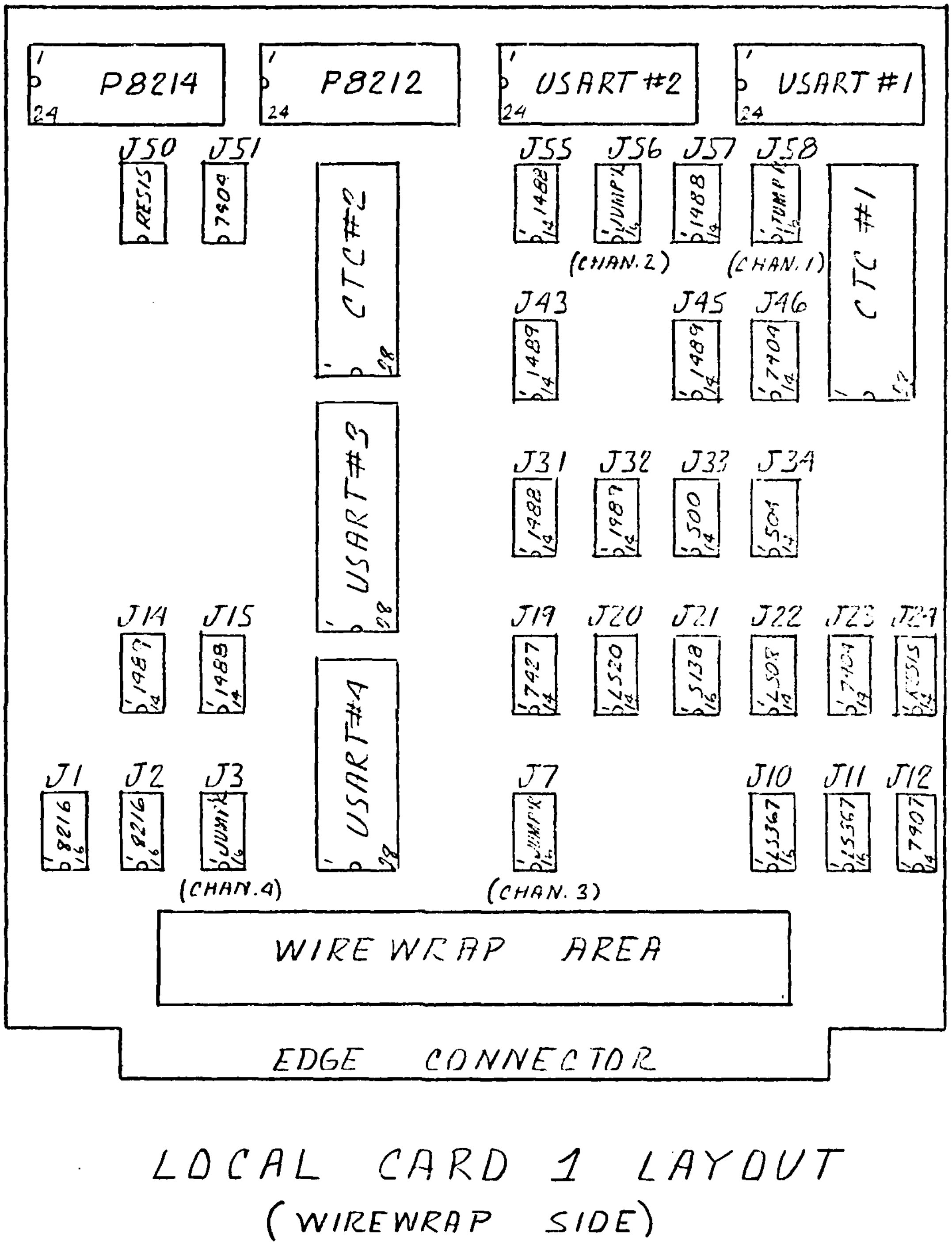


Fig A-3a Local Card 1 Layout.

AD-A115 671 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 17/2
PROTOTYPE DELNET USING THE UNIVERSAL NETWORK INTERFACE DEVICE.(U)
DEC 81 C E PAPP
UNCLASSIFIED AFIT/GE/EE/81D-46 NL

2 OF 2
AD A
115 671

END
DATE
FILMED
07-82
DTIC

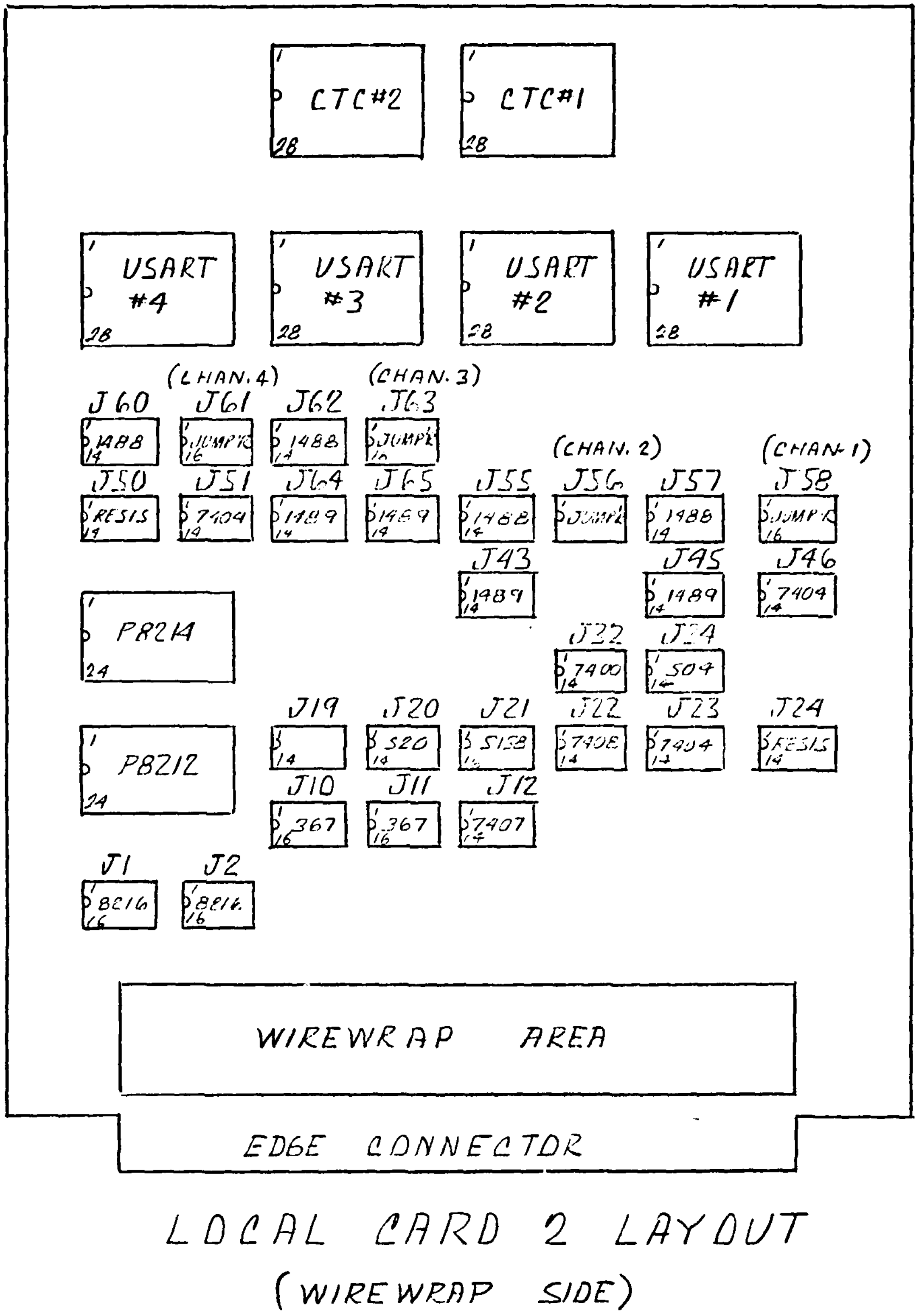
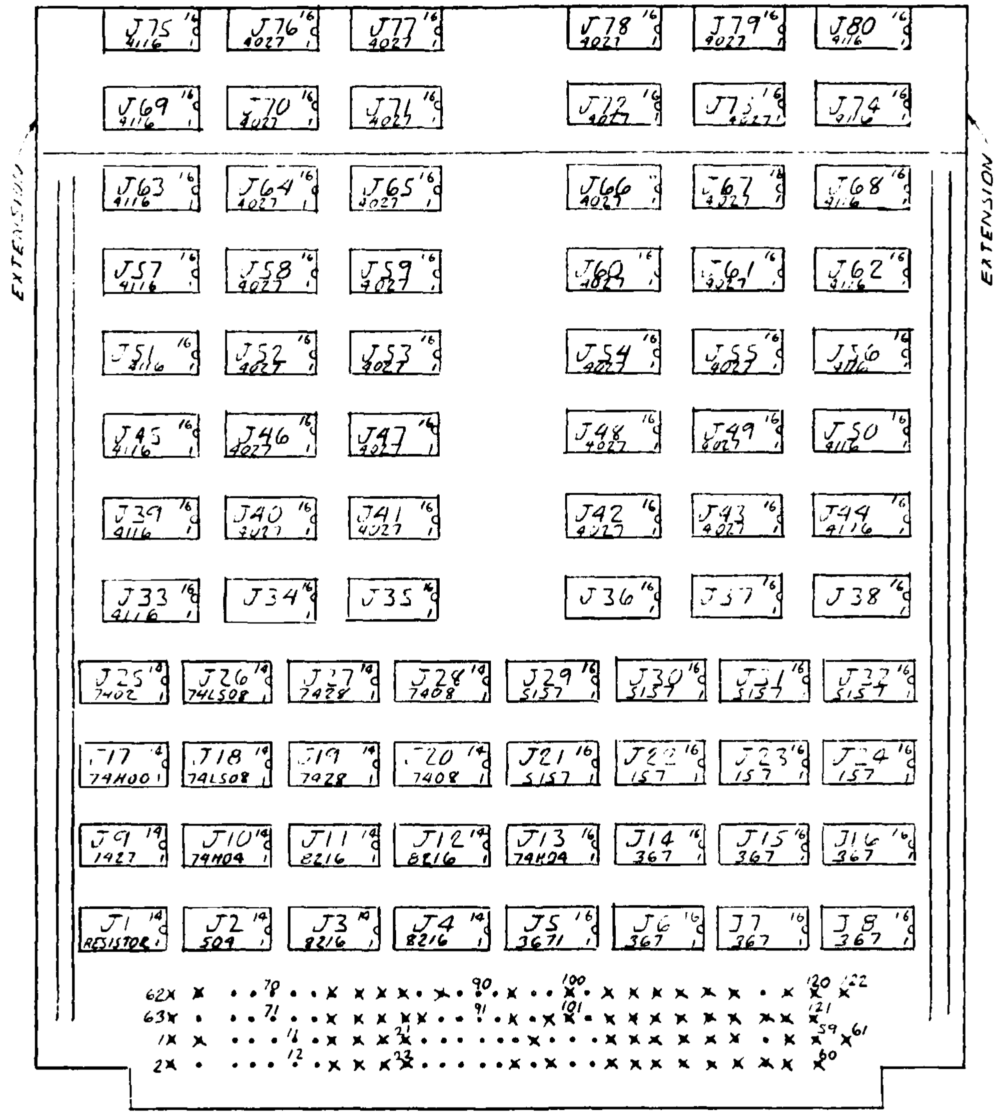


Fig A-3b Local Card 2 Layout.



SYSTEM MEMORY CARD

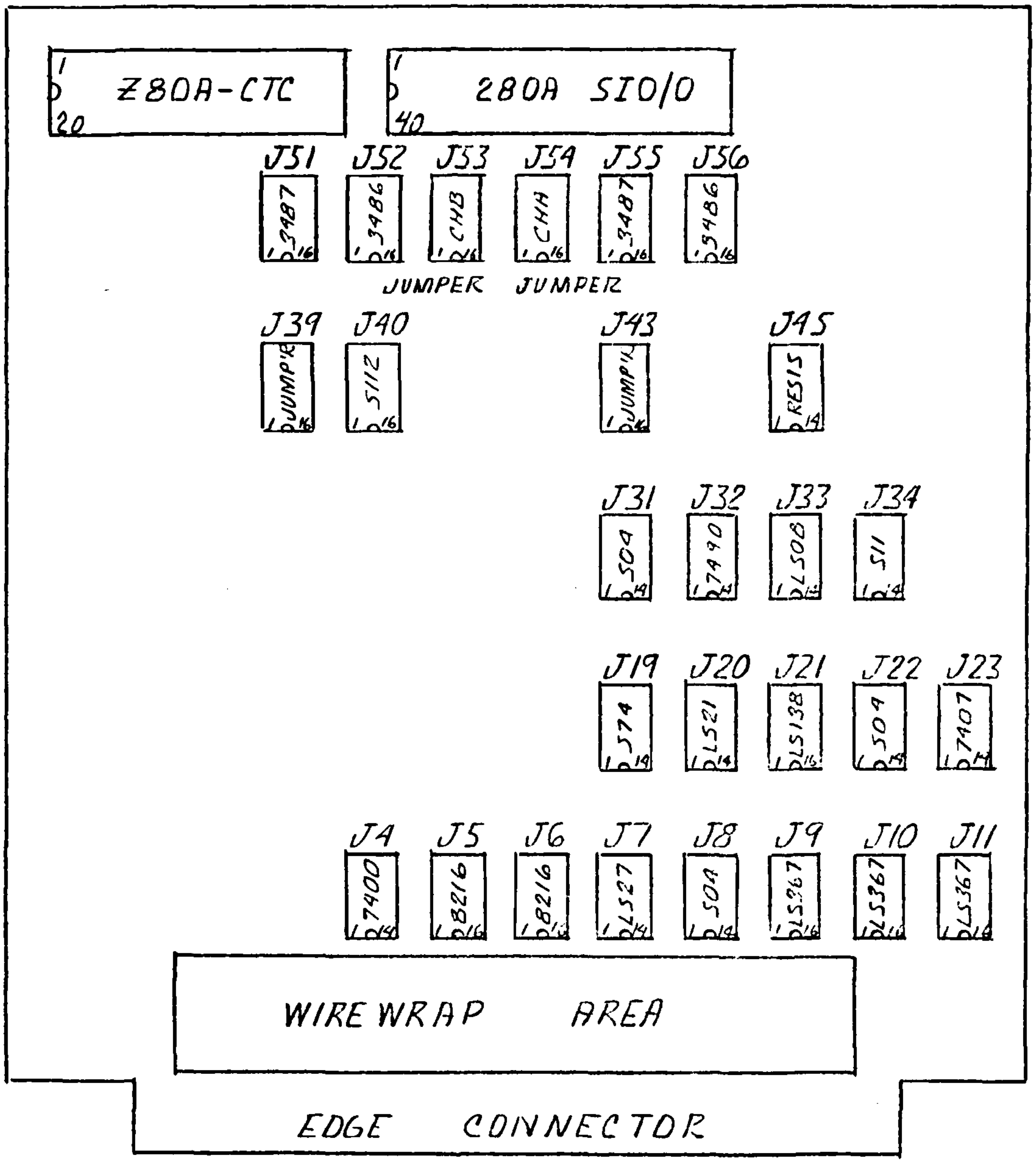
(VIEW FROM WIREWRAP SIDE)

(UNID I & 2)

✕ PINS INSTALLED

• NO PINS INSTALLED

Fig A-4 System Memory Card Layout.



NETWORK CARD 1 AND 2 LAYOUT
(WIREWRAP SIDE)

Fig A-5 Network Card Layout.

Appendix B

Circuit Card Schematics

This appendix contains the circuit schematics for five of the UNID circuit cards, including the motherboard. The circuit schematics for the local processor and the network processor are contained in the Z80 MCB User's Manual (15).

This figure lists the wiring connections made on the motherboards. The head of each column corresponds to a connector on the motherboard. The verticle column at the left of the figure lists the particular signals being described. The numbers in the figure represent the wirewrap pin numbers on the given connector where the signal is attached. All signals on the motherboard are grouped according to function.

	J2	J3	J1	J5	J7	J9	J15	J16	J17	J18	J19	J20	J21
LOCAL PROCESSOR ADDRESS BUS (X)													
XA0	*	103	38	90	*	103	*	*	*	*	*	*	*
XA1	*	102	43	92	*	102	*	*	*	*	*	*	*
XA2	*	101	44	94	*	101	*	*	*	*	*	*	*
XA3	*	100	45	96	*	100	*	*	*	*	*	*	*
XA4	*	98	46	91	*	98	*	*	*	*	*	*	*
XA5	*	29	47	93	*	29	*	*	*	*	*	*	*
XA6	*	30	48	95	*	30	*	*	*	*	*	*	*
XA7	*	26	49	97	*	26	*	*	*	*	*	*	*
XA8	*	27	50	29	*	27	*	*	*	*	*	*	*
XA9	*	89	51	31	*	89	*	*	*	*	*	*	*
XA10	*	91	52	33	*	91	*	*	*	*	*	*	*
XA11	*	37	53	35	*	37	*	*	*	*	*	*	*
XA12	*	97	54	30	*	97	*	*	*	*	*	*	*
XA13	*	36	56	32	*	36	*	*	*	*	*	*	*
XA14	*	94	57	34	*	94	*	*	*	*	*	*	*
XA15	*	32	58	36	*	32	*	*	*	*	*	*	*
LOCAL PROCESSOR DATA BUS (X)													
XD0	*	13	15	68	*	13	*	*	*	*	*	*	*
XD1	*	75	16	69	*	75	*	*	*	*	*	*	*
XD2	*	71	17	7	*	71	*	*	*	*	*	*	*
XD3	*	8	18	8	*	8	*	*	*	*	*	*	*
XD4	*	68	19	70	*	68	*	*	*	*	*	*	*
XD5	*	5	20	71	*	5	*	*	*	*	*	*	*
XD6	*	12	21	9	*	12	*	*	*	*	*	*	*
XD7	*	73	22	10	*	73	*	*	*	*	*	*	*

Fig B-1 Motherboard Wiring List.

	J2	J3	J1	J5	J7	J9	J15	J16	J17	J18	J19	J20	J21
LOCAL PROCESSOR CONTROL SIGNALS (X)													
XWR	*	23	24	117	*	23	*	*	*	*	*	*	*
XRFSH	*	35	95	110	*	*	*	*	*	*	*	*	*
XRD	*	116	39	111	*	116	*	*	*	*	*	*	*
XM1	*	115	*	*	*	115	*	*	*	*	*	*	*
XMREQ	*	85	35	110	*	*	*	*	*	*	*	*	*
XIORQ	*	4	*	*	*	4	*	*	*	*	*	*	*
XWAIT	*	119	*	55	*	*	*	*	*	*	*	*	*
XINT	*	79	*	*	*	79	*	*	*	*	*	*	*
XRESET	*	10	*	*	*	31	*	*	*	*	*	5	*
XCLK	39	99	*	*	*	99	*	*	*	*	*	*	*
XCLK/2	*	118	*	56	*	*	*	*	*	*	*	*	*
LOCAL PROCESSOR MONITOR CHANNEL (X)													
XTxD	*	15	*	*	*	*	*	*	*	2	*	*	*
XRxD	*	7	*	*	*	*	*	*	*	3	*	*	*
XRTS	*	14	*	*	*	*	*	*	*	4	*	*	*
XCTS	*	11	*	*	*	*	*	*	*	5	*	*	*
XDSR	*	74	*	*	*	*	*	*	*	6	*	*	*
XGND	*	64	*	*	*	*	*	*	*	7	*	*	*
XLSD	*	80	*	*	*	*	*	*	*	8	*	*	*
XDTR	*	76	*	*	*	*	*	*	*	20	*	*	*
NETWORK PROCESSOR ADDRESS BUS (Y)													
YA0	103	*	99	100	103	*	*	*	*	*	*	*	*
YA1	102	*	104	102	102	*	*	*	*	*	*	*	*
YA2	101	*	105	104	101	*	*	*	*	*	*	*	*
YA3	100	*	106	106	100	*	*	*	*	*	*	*	*
YA4	98	*	107	101	98	*	*	*	*	*	*	*	*
YA5	29	*	108	103	29	*	*	*	*	*	*	*	*
YA6	30	*	109	105	30	*	*	*	*	*	*	*	*
YA7	26	*	110	107	26	*	*	*	*	*	*	*	*
YA8	27	*	111	39	*	*	*	*	*	*	*	*	*
YA9	89	*	112	41	*	*	*	*	*	*	*	*	*
YA10	91	*	113	43	*	*	*	*	*	*	*	*	*
YA11	37	*	114	45	*	*	*	*	*	*	*	*	*
YA12	97	*	115	40	*	*	*	*	*	*	*	*	*
YA13	36	*	117	42	*	*	*	*	*	*	*	*	*
YA14	94	*	118	44	*	*	*	*	*	*	*	*	*
YA15	32	*	119	46	*	*	*	*	*	*	*	*	*
NETWORK PROCESSOR DATA BUS (Y)													
YD0	13	*	76	74	13	*	*	*	*	*	*	*	*
YD1	75	*	77	75	75	*	*	*	*	*	*	*	*

Fig B-1 Motherboard Wiring List.

	J2	J3	J1	J5	J7	J9	J15	J16	J17	J18	J19	J20	J21
YD2	71	*	78	13	71	*	*	*	*	*	*	*	*
YD3	8	*	79	14	8	*	*	*	*	*	*	*	*
YD4	68	*	80	76	68	*	*	*	*	*	*	*	*
YD5	5	*	81	77	5	*	*	*	*	*	*	*	*
YD6	12	*	82	15	12	*	*	*	*	*	*	*	*
YD7	73	*	83	16	73	*	*	*	*	*	*	*	*
NETWORK PROCESSOR CONTROL SIGNALS (Y)													
YWR	23	*	25	58	23	*	*	*	*	*	*	*	*
YRFSH	35	*	34	51	*	*	*	*	*	*	*	*	*
YRD	116	*	40	113	116	*	*	*	*	*	*	*	*
YWAIT	119	*	*	118	119	*	*	*	*	*	*	*	*
YM1	115	*	*	*	115	*	*	*	*	*	*	*	*
YMRQ	85	*	96	112	85	*	*	*	*	*	*	*	*
YIORQ	4	*	*	*	4	*	*	*	*	*	*	*	*
YINT	79	*	*	*	79	*	*	*	*	*	*	*	*
YRESET	10	*	*	*	31	*	*	*	*	*	*	*	10
YCLK	99	*	*	*	99	*	*	*	*	*	*	*	*
YCLK/2	118	*	*	*	58	*	*	*	*	*	*	*	*
NETWORK PROCESSOR MONITOR CHANNEL (Y)													
YTxD	15	*	*	*	*	*	*	2	*	*	*	*	*
YRxD	7	*	*	*	*	*	*	3	*	*	*	*	*
YRTS	14	*	*	*	*	*	*	4	*	*	*	*	*
YCTS	11	*	*	*	*	*	*	5	*	*	*	*	*
YDSR	74	*	*	*	*	*	*	6	*	*	*	*	*
YGND	64	*	*	*	*	*	*	7	*	*	*	*	*
YLSD	80	*	*	*	*	*	*	8	*	*	*	*	*
YDTR	76	*	*	*	*	*	*	20	*	*	*	*	*
LOCAL CHANNEL 1													
TxD	*	*	*	*	*	57	*	*	*	*	*	*	2
RxD	*	*	*	*	*	58	*	*	*	*	*	*	3
RTS	*	*	*	*	*	52	*	*	*	*	*	*	4
CTS	*	*	*	*	*	53	*	*	*	*	*	*	5
DSR	*	*	*	*	*	54	*	*	*	*	*	*	6
LSD	*	*	*	*	*	56	*	*	*	*	*	*	8
DTR	*	*	*	*	*	55	*	*	*	*	*	*	20
LOCAL CHANNEL 2													
TxD	*	*	*	*	*	49	*	*	*	*	2	*	*
RxD	*	*	*	*	*	50	*	*	*	*	3	*	*
RTS	*	*	*	*	*	44	*	*	*	*	4	*	*
CTS	*	*	*	*	*	45	*	*	*	*	5	*	*

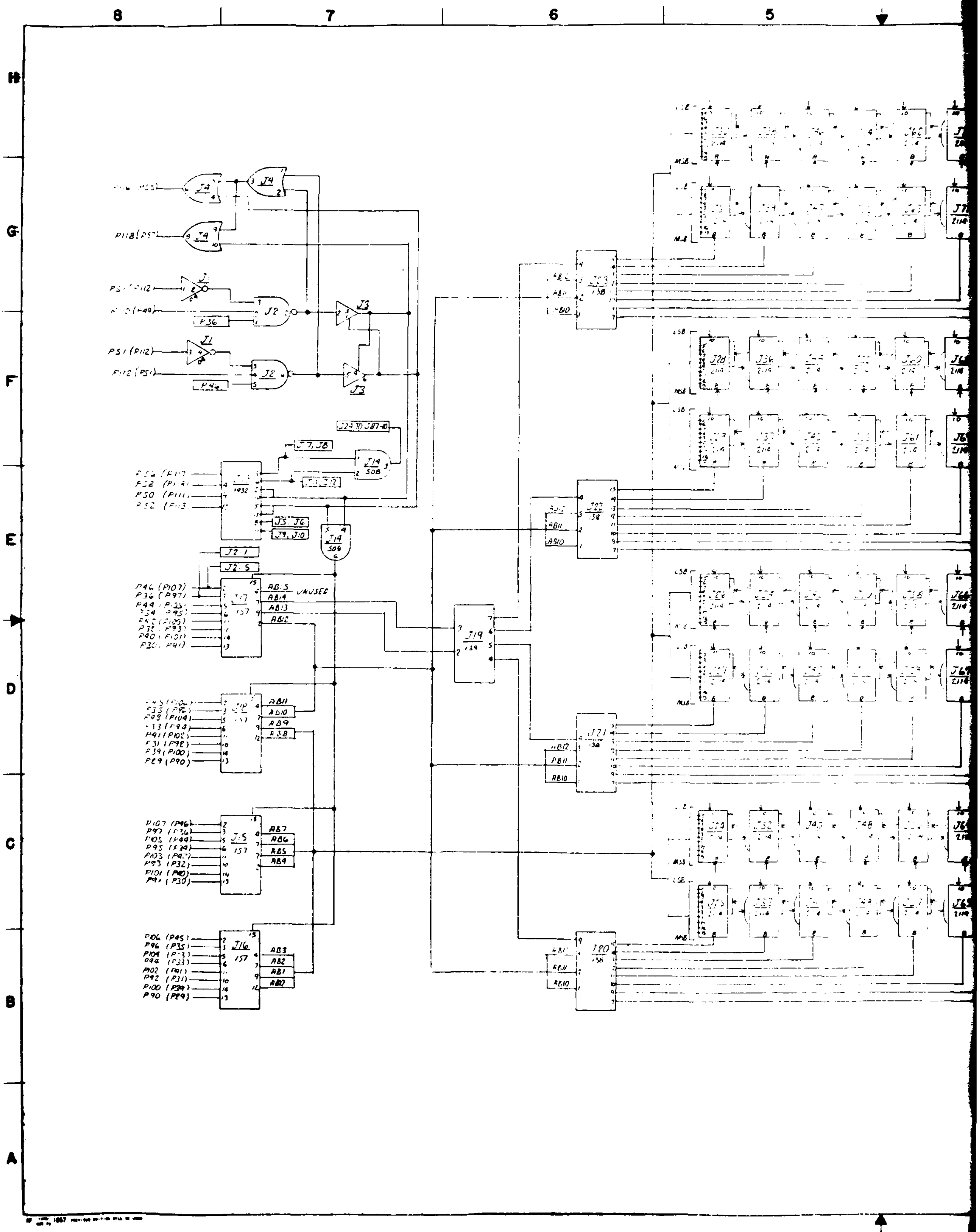
Fig B-1 Motherboard Wiring List.

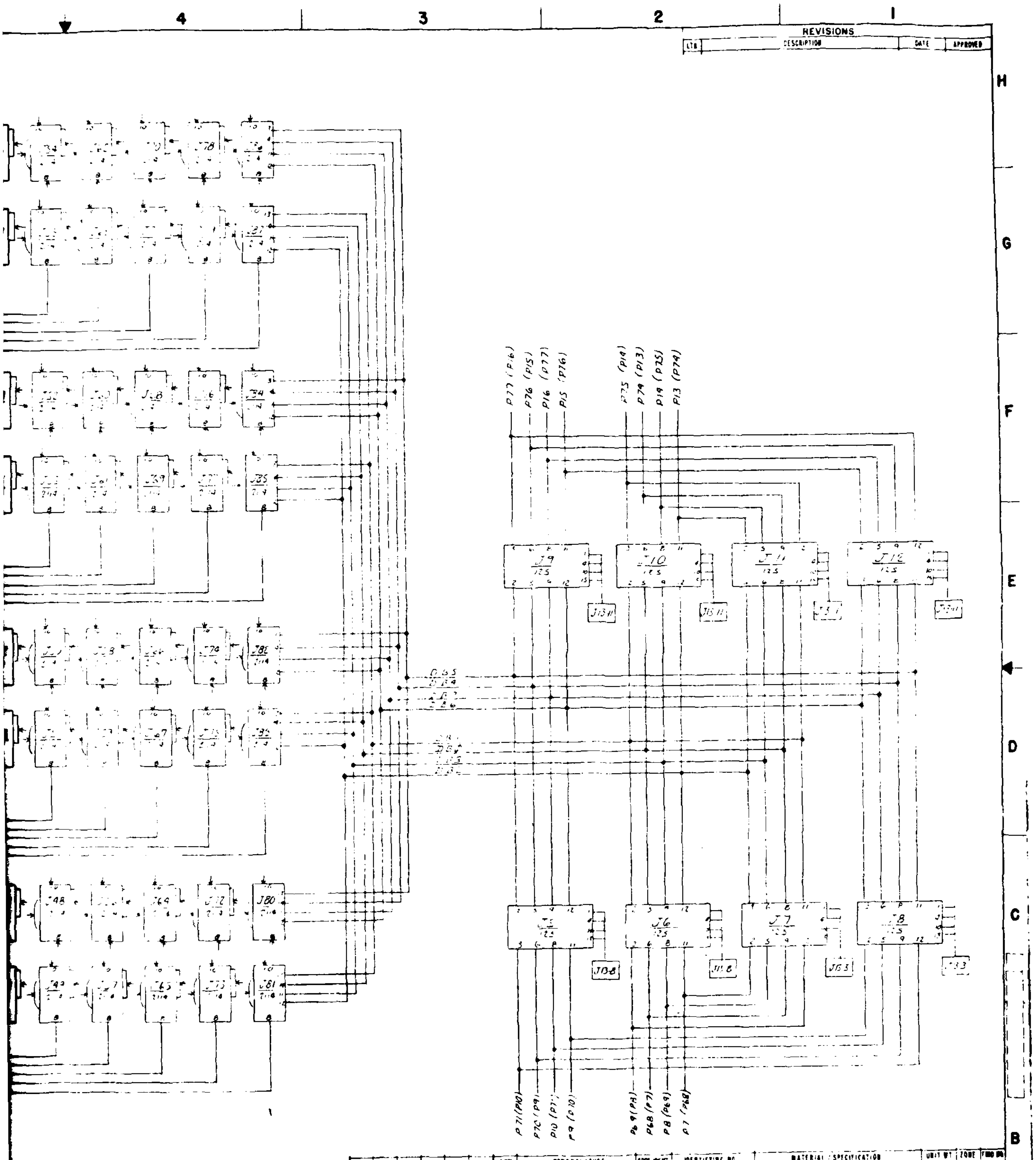
	J2	J3	J1	J5	J7	J9	J15	J16	J17	J18	J19	J20	J21
DSR	*	*	*	*	*	46	*	*	*	*	6	*	*
LSD	*	*	*	*	*	48	*	*	*	*	8	*	*
DTR	*	*	*	*	*	47	*	*	*	*	20	*	*
LOCAL CHANNEL 3													
TxD	*	*	*	*	*	37	*	*	2	*	*	*	*
RxD	*	*	*	*	*	38	*	*	3	*	*	*	*
RTS	*	*	*	*	*	32	*	*	4	*	*	*	*
CTS	*	*	*	*	*	33	*	*	5	*	*	*	*
DSR	*	*	*	*	*	34	*	*	6	*	*	*	*
LSD	*	*	*	*	*	36	*	*	8	*	*	*	*
DTR	*	*	*	*	*	35	*	*	20	*	*	*	*
LOCAL CHANNEL 4													
TxD	*	*	*	*	*	19	2	*	*	*	*	*	*
RxD	*	*	*	*	*	20	3	*	*	*	*	*	*
RTS	*	*	*	*	*	14	4	*	*	*	*	*	*
CTS	*	*	*	*	*	15	5	*	*	*	*	*	*
DSR	*	*	*	*	*	16	6	*	*	*	*	*	*
LSD	*	*	*	*	*	18	8	*	*	*	*	*	*
DTR	*	*	*	*	*	17	20	*	*	*	*	*	*

In addition to the wiring connections listed above, each MCB processor has a number of jumper connections on the edge connector. See the DELNET User's Manual, Appendix G, or the Z80 MCB User's Manual (15) for further information.

Fig B-1 Motherboard Wiring List.

Fig B-2 Shared Memory 1 and 2 Schematic.


8
7
6
5
H
G
F
E
D
C
B
A
P116 (P55)
J4
J4
P118 (P57)
J4
PS1 (P112)
J1
P110 (P49)
J2
P36
PS1 (P112)
J1
P112 (PS1)
J2
P46
J3
J3
J24 TO J87-10
J7, J8
J14
508
1932
P54 (P117)
P52 (P114)
P50 (P111)
P52 (P113)
J5, J6
J9, J10
J14
508
J2-1
J2-5
P46 (P107)
P36 (P97)
P44 (P105)
P34 (P95)
P42 (P103)
P32 (P93)
P40 (P101)
P30 (P91)
J17
157
AB15 UNUSED
AB14
AB13
AB12
J19
139
P45 (P106)
P35 (P96)
P43 (P104)
P33 (P94)
P41 (P102)
P31 (P92)
P39 (P100)
P29 (P90)
J18
157
AB11
AB10
AB9
AB8
P107 (P46)
P97 (P36)
P105 (P44)
P95 (P34)
P103 (P42)
P93 (P32)
P101 (P40)
P91 (P30)
J15
157
AB7
AB6
AB5
AB4
P106 (P45)
P96 (P35)
P104 (P43)
P94 (P33)
P102 (P41)
P92 (P31)
P100 (P39)
P90 (P29)
J16
157
AB3
AB2
AB1
AB0
J23
138
AB12
AB11
AB10
J22
138
AB12
AB11
AB10
J21
138
AB12
PB11
AB10
J20
138
AB12
AB11
AB10
LSB
MSB
J28
2114
J36
2114
J29
2114
J37
2114
J61
2114
J62
2114
J26
2114
J27
2114
J23
J32
2114
J43
J48
J65
2114
J66
2114
J67
2114
J68
2114
J69
2114
J71
2114



SYM	NOMENCLATURE	CODE IDENT	IDENTIFYING NO	MATERIAL / SPECIFICATION	UNIT WT	ZONE	FIND NO

QUANTITY REQUIRED PER DASH NO

PARTS LIST

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES – TOLERANCES ON FRACTIONS DECIMALS ANGLES ± .XX ± .XXX ±

DR

CHK

ENG

AIR FORCE AUTHENTICATION

AIR FORCE RELEASE

U.S. AIR FORCE

TITLE

SIZE AIR FORCE CODE IDENT NO DRAWING NO

F

SCALE

SHEET

CURRENT CODE IDENT NO

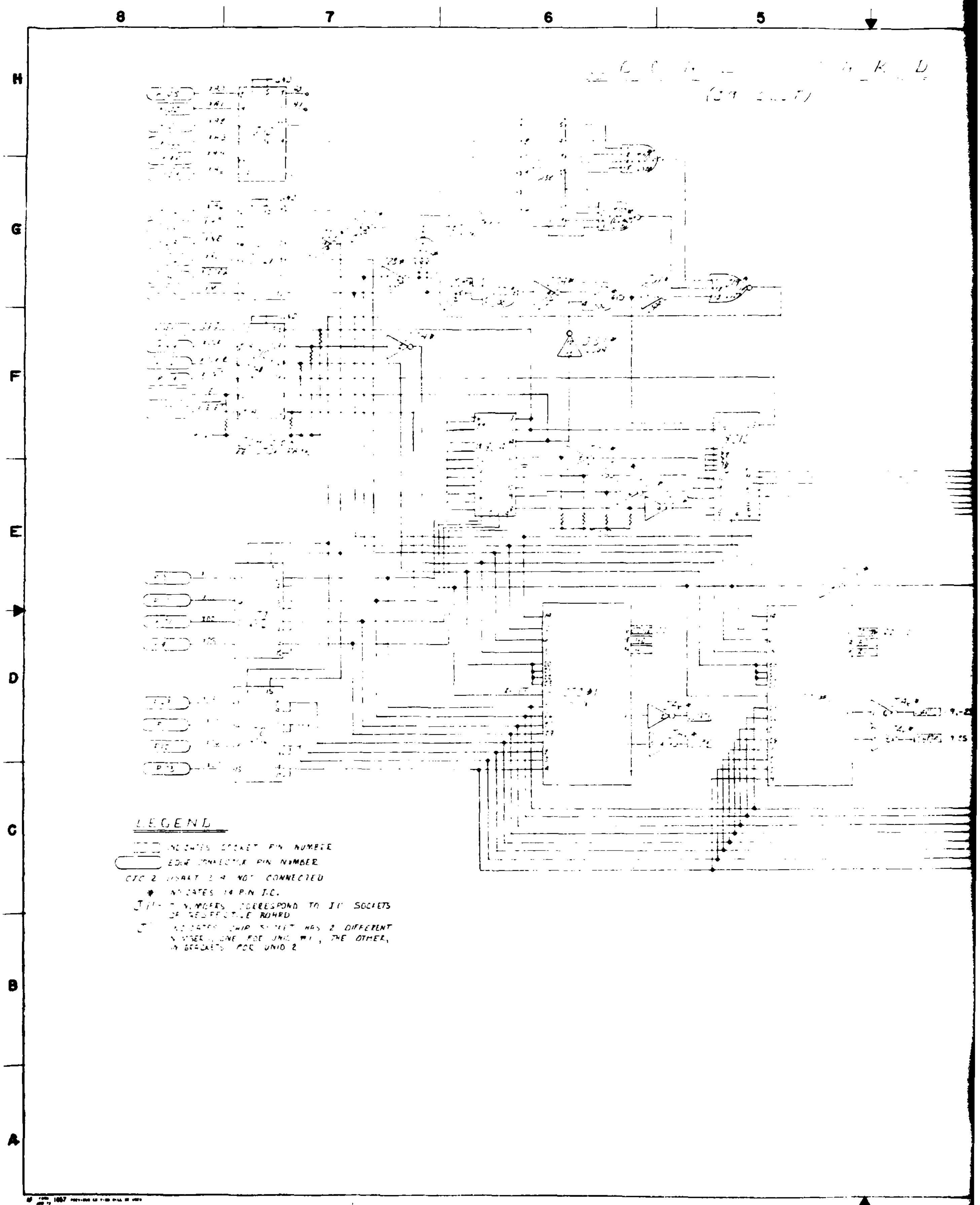
NEXT ASSEMBLY

USED ON

APPLICATION

A

Fig B-3 Local Card 1 and 2 Schematic.



LEGEND

INDICATES SOCKET PIN NUMBER

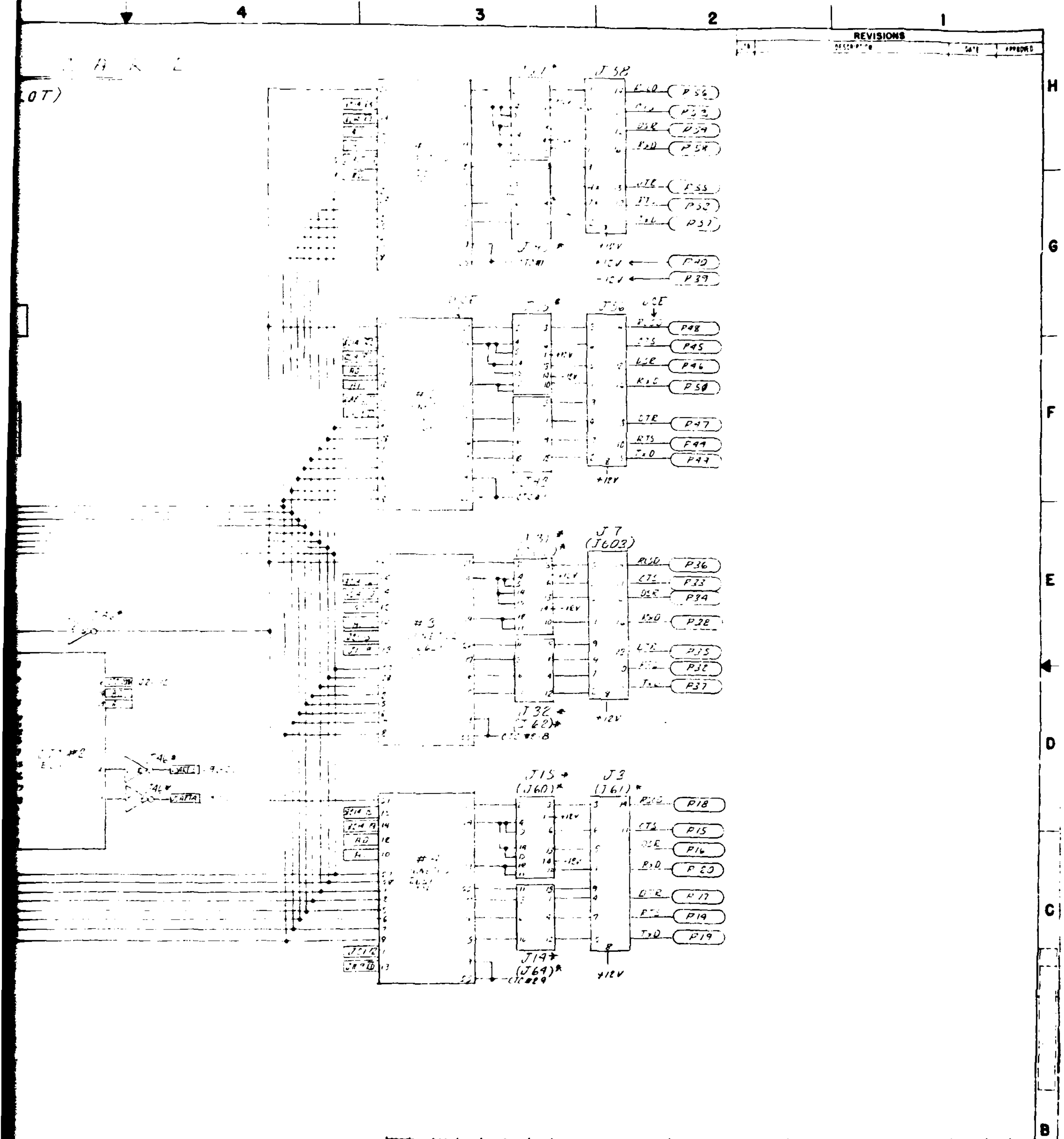
EDGE CONNECTOR PIN NUMBER

CTC 2 INSERT 3-4 NOT CONNECTED

✱ INDICATES 14 PIN I.C.

J1/1- NUMBERS CORRESPOND TO IC SOCKETS OF RESPECTIVE BOARD

INDICATES CHIP SOCKET HAS 2 DIFFERENT NUMBERS, ONE FOR UNIT #1, THE OTHER, IN BRACKETS FOR UNIT 2



SYM	NOMENCLATURE	CODE IDENT	IDENTIFYING NO	MATERIAL / SPECIFICATION	UNIT WT	ZONE	FIND NO

QUANTITY REQUIRED PER DASH NO

PARTS LIST

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES - TOLERANCES ON: FRACTIONS DECIMALS ANGLES
± .XX ± .XXX ±

U S AIR FORCE

TITLE

DR
CHK
ENG
AIR FORCE AUTHENTICATION
AIR FORCE OF CASE

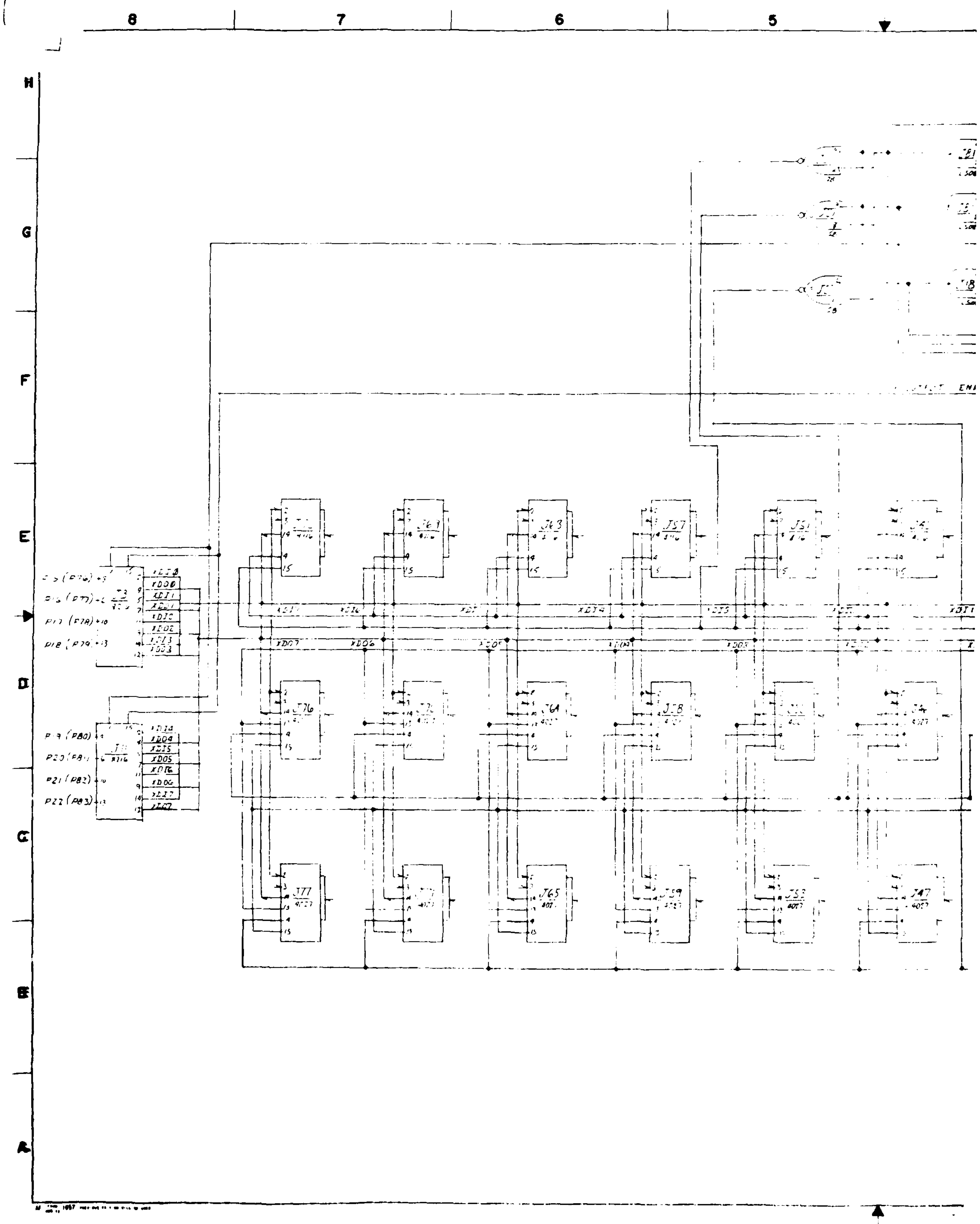
SIZE AIR FORCE CODE IDENT NO. DRAWING NO.
F

SCALE

CURRENT CODE IDENT NO
NEXT ASSEMBLY USED ON
APPLICATION

A

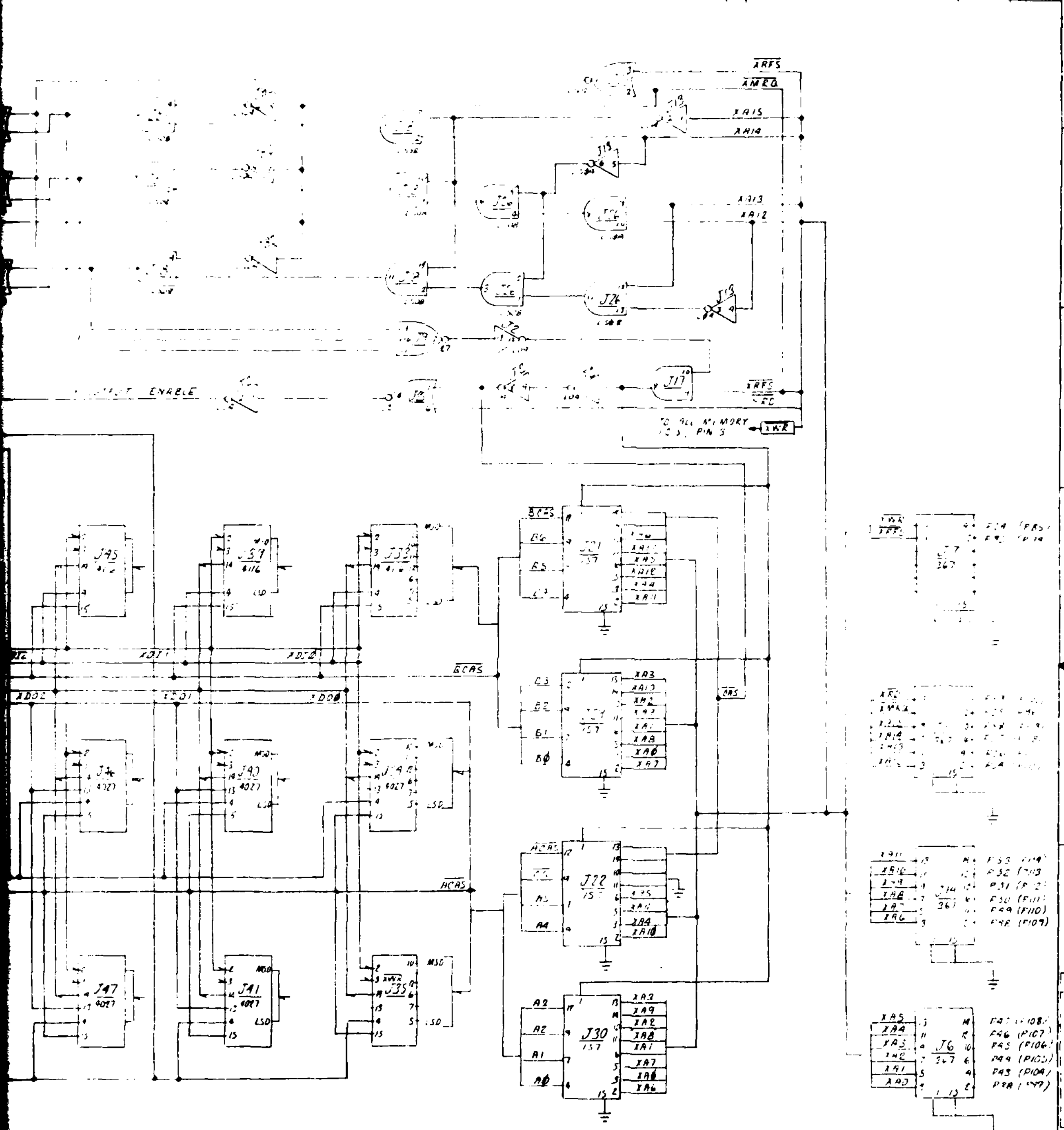
Fig B-4a System Memory 1 and 2 Schematic (Local Processor).

8
7
6
5
H
G
F
E
D
C
B
A
P15 (P76)
P16 (P77)
P17 (P78)
P18 (P79)
J3
4116
XDI0
XDO0
XDI1
XDO1
XDI2
XDO2
XDI3
XDO3
P19 (P80)
P20 (P81)
P21 (P82)
P22 (P83)
J11
4116
XDI4
XDO4
XDI5
XDO5
XDI6
XDO6
XDI7
XDO7
J69
4116
J63
J57
4116
J51
4116
J76
4027
J64
4027
J58
4027
J46
4027
J77
4027
J65
4027
J59
4027
J53
4027
J47
4027
J28
J18
OUTPUT ENA

2

REVISIONS

LTR	DESCRIPTION	DATE	APPROVED



						SYM	NOMENCLATURE	CODE IDENT	IDENTIFYING NO	MATERIAL / SPECIFICATION	UNIT WT	ZONE	FIND NO

QUANTITY REQUIRED PER DASH NO

PARTS LIST

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES - TOLERANCES ON
FRACTIONS DECIMALS ANGLES

U.S. AIR FORCE

TITLE

SIZE CODE IDENT NO DRAWING NO
F

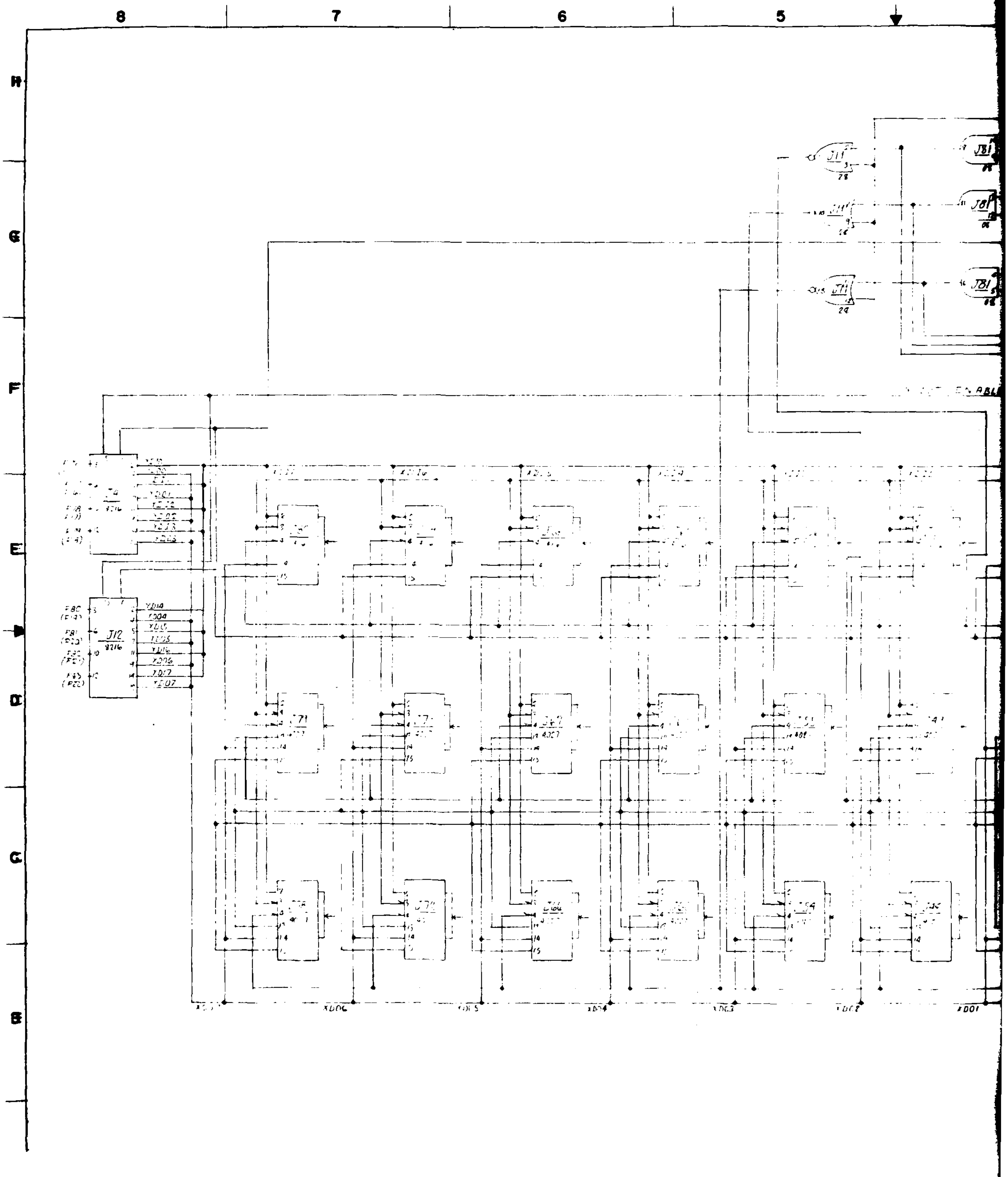
CURRENT CODE IDENT NO

NEXT ASSEMBLY USED ON

APPLICATION

A

Fig B-4b System Memory 1 and 2 Schematic (Network Processor).


8
7
6
5
H
G
F
E
D
C
B
J11
J81
J11
J81
J11
J81
ENABL
J4
8216
J12
8216
F8C
F81
XD14
XD04
XD15
XD05
XD16
XD06
XD17
XD07
XD00
XD01
XD02
XD03
J71
4027
J67
4027
XD06
XD05
XD04
XD03
XD02
XD01

4 3 2 1

REVISIONS

LTR	DESCRIPTION	DATE	APPROVED

H

G

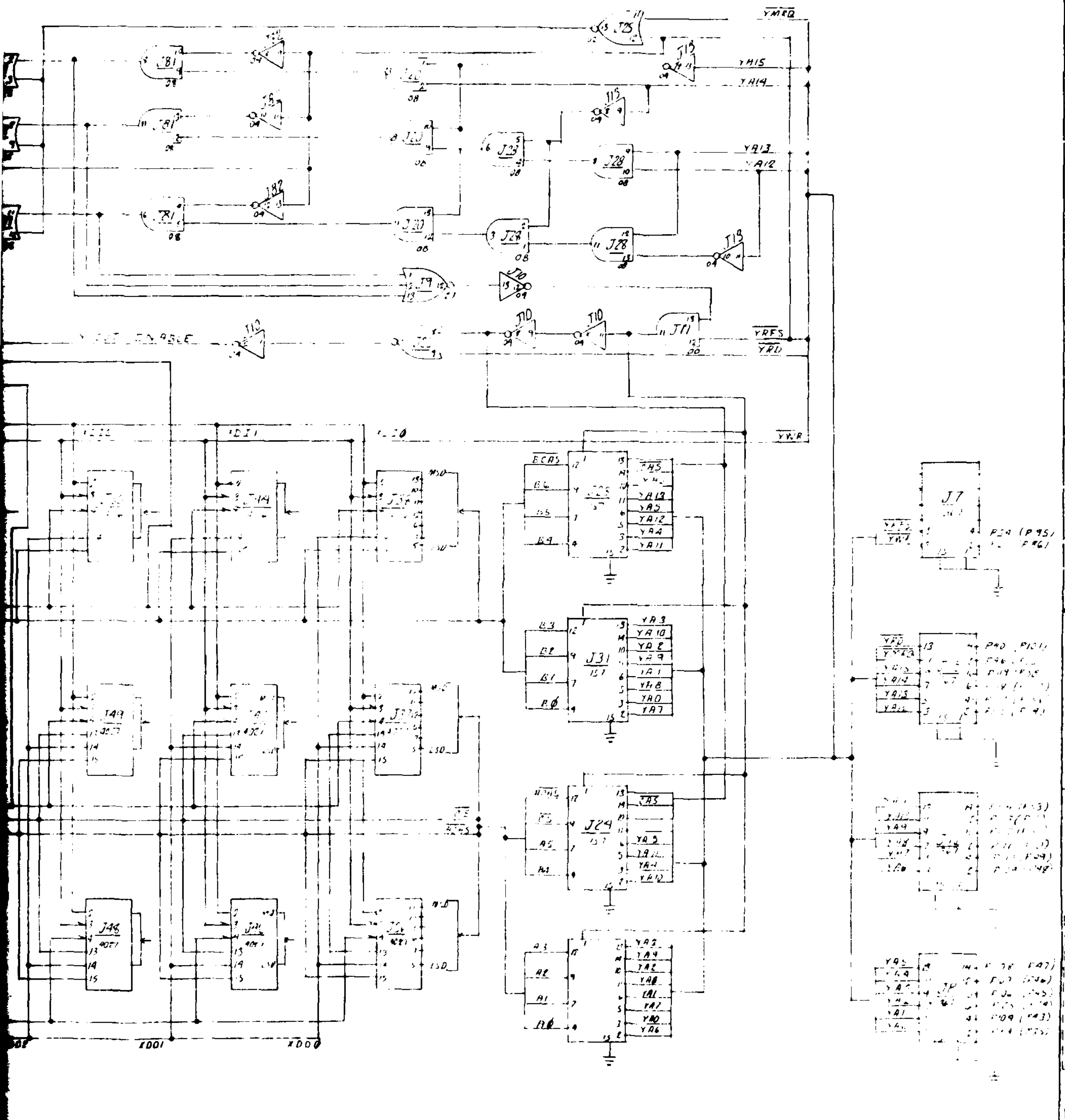
F

E

D

C

B



SYM	NOMENCLATURE	CODE IDENT	IDENTIFYING NO	MATERIAL SPECIFICATION	UNIT WT	ZONE

QUANTITY REQUIRED PER DASH NO

PARTS LIST

UNLESS OTHERWISE SPECIFIED
DIMENSIONS ARE IN INCHES - TOLERANCES ON:
FRACTIONS DECIMALS ANGLES
± .XX± .XXX± ±

DR
CHK
ENG
AIR FORCE AUTHENTICATION
AIR FORCE RELEASE

U.S. AIR FORCE

TITLE

SIZE F AIR FORCE DRAWING NO

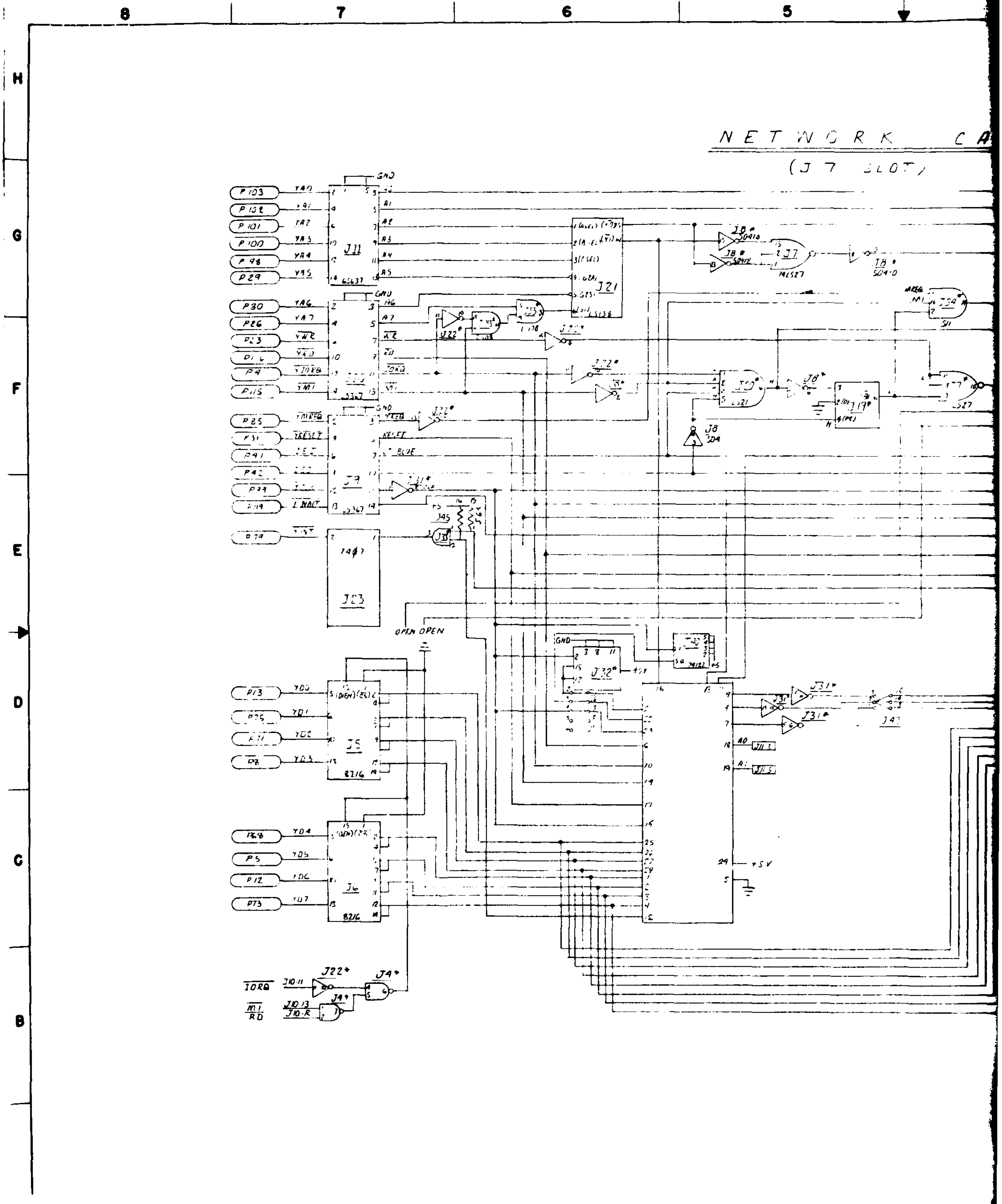
SCALE SHEET

CONTRACT CODE IDENT NO
NEXT ASSEMBLY USED ON
APPLICATION

Fig B-5 Network Card 1 and 2 Schematic.

NETWORK CA

(J7 SLOT)

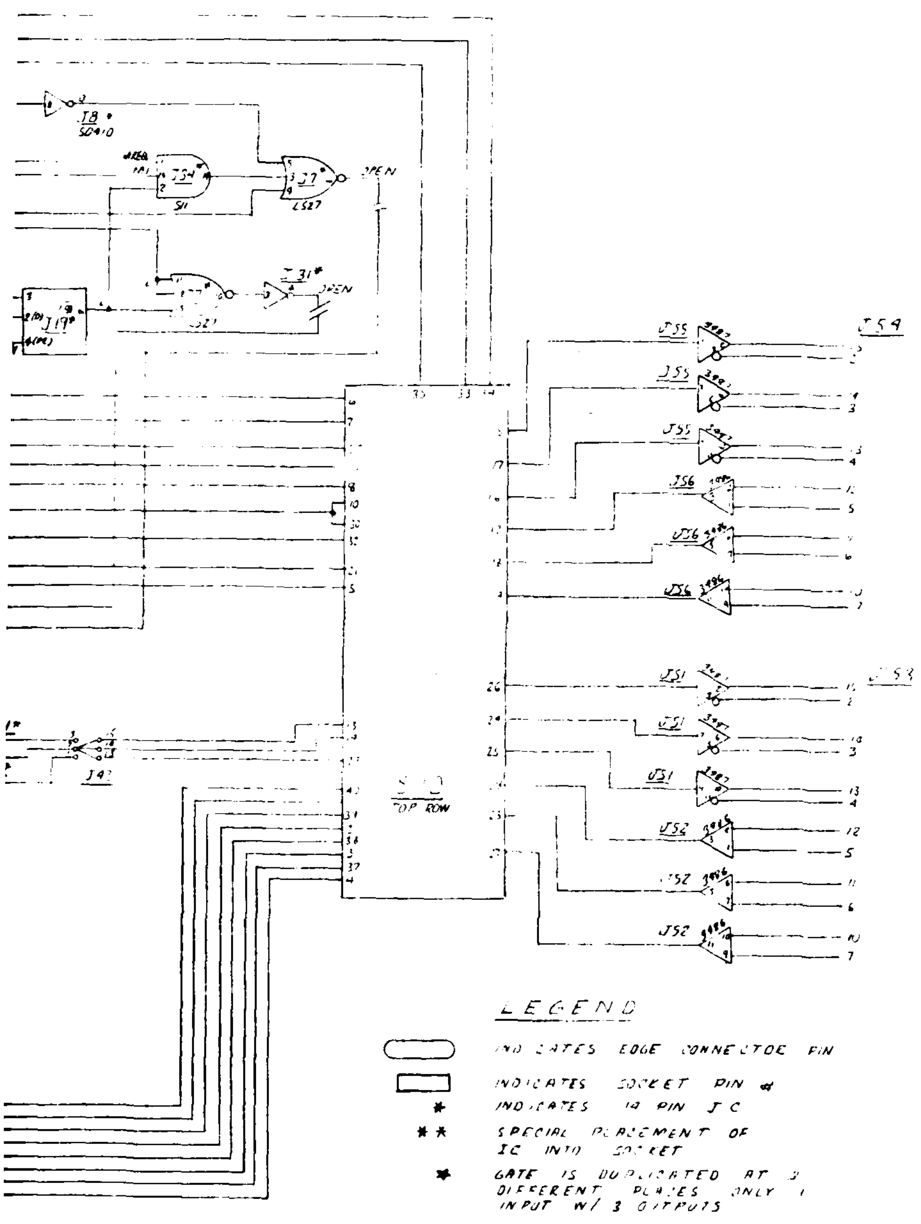


4 3 2 1

REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED

ORK CARD

7 SLOT)



LEGEND

⬭ INDICATES EDGE CONNECTOE PIN

▭ INDICATES SOCKET PIN #

* INDICATES 14 PIN IC

** SPECIAL PLACEMENT OF IC INTO SOCKET

✱ GATE IS DUPLICATED AT 3 DIFFERENT PLACES ONLY 1 INPUT W/ 3 OUTPUTS

QUANTITY REQUIRED PER DASH NO | SYM | NOMENCLATURE | CODE IDENT | IDENTIFYING NO | MATERIAL / SPECIFICATION | UNIT WT | ZONE | FIND NO

PARTS LIST

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES TOLERANCES ON FRACTIONS DECIMALS ANGLES

U S AIR FORCE

TITLE

DR

CHK

ENG

AIR FORCE AUTHENTICATION

AIR FORCE RELEASE

SIZE F

SCALE

CURRENT CODE IDENT NO | NEXT ASSEMBLY | USED ON

APPLICATION

H G F E D C B A

Appendix C

EPROM Programming Using the MCZ1/25 System.

The instructions for EPROM programming are contained under seperate cover. See Zilog's *Z-80 PPB Hardware Reference Manual* and *Z80-PPB Hardware User's Manual*. These publications are published by Zilog, Inc., Cupertino, California. The instructions contained in this appendix are local modifications to the instructions contained in the above Zilog publications.

The Zilog PROM Programmer Board (PPB) provides the MCZ1/25 system with the programming capability for the following PROMs:

Harris Bipolar (16 and 24 pin models)

7610, 7611, 7620, 7621, 7640, 7641

EPROM (24 pin models)

2704, 2708

The PPB card is inserted, with the extender card, in card slot J9 inside the MCZ1/25. The PPB card is installed with the component side of the board facing away from the disk drives contained in the MCZ1/25. The extender card is used to allow easy access to the zero-force PROM IC sockets at the top of the PPB card. Pin 1 on each of the three zero-force sockets is in the upper right corner, looking at the component side of the PPB card.

Use of the PPB card requires a special utility program called ZPROG. The program is cataloged under the file ZPROG.H2708.MCZ stored on an RIO Utilities disk titled "MCZ RIO Utilities + More". This utility program permits the user to program, verify, copy, list, and duplicate PROMs.

Once the PPB disk is installed, the program is executed by typing the following:

```
ZPROG.H2708.MCZ
```

The program will return the "pound sign" as a prompt during the PPB sesion. Select the PPB option desired by typing in one of the following:

```
FILE * filename
LIST *
PROGRAM filename *
DUPLICATE *
```

The "*" must be included with each of these commands. The program will continue in each case with a series of questions. The FILE command copies the contents of the PROM and places the information in the file specified by "filename" as a procedure file (file type P). The LIST command copies the contents of a PROM and displays the information on the video console. The PROGRAM command programs a PROM with the contents of the file specified by "filename". The DUPLICATE command can copy (or verify) the contents of one PROM to another PROM. More detailed information may be found in the Zilog publications listed above.

Appendix D

Local and Network Processor Monitor Listing.

The op code listing is contained in the DELNET User's Manual, Appendix G. An assembly language listing is available under seperate cover. The assembly language listing is not published.

Appendix E

Monitor Command Language Grammar.

This appendix contains the grammar and description for each command currently available in the local and network processor monitors.

In the following command descriptions, angle brackets "< >" are used to enclose descriptive names for the quantities to be entered, and are not actually to be entered. Brackets "[ ]" enclose optional quantities. Each command listed below may be entered by typing the whole name or the first letter. The following are the commands currently implemented in the UNID monitors:

DISPLAY <hex address> [<number of bytes to be displayed>]
Displays on the terminal the contents of memory locations starting at the given address, for the given number of bytes. If no number of bytes is given, memory locations will be displayed one at a time with an opportunity to change the contents of each byte. In this mode each byte is displayed with its address. The order is address followed by contents followed by a space. If it is desired to change the contents of that location, type the new byte in following the space. A carriage return will enter the new byte and retype the line for verification. A line feed without typing in a new byte will cause the next sequential byte to be typed out. A carriage return without typing a new byte will exit the command. If more than one byte is displayed, the contents of the designated memory locations are displayed in both hex notation and as ASCII characters.

FILL <beginning address> <ending address> <data>
Fills the memory locations between the beginning and ending addresses, inclusive, with the single byte of data

specified. This command will not affect ROM or areas of non-existent memory.

JUMP <hex address>

Branches unconditionally to the given address to execute a user's program.

LOAD <filename>

Loads an RIO file specified by the filename into the UNID memory through Local Channel 1 from the MCZ1/25 I/O Channel (J13). Before this command is given, the command DO SETUP.UNID must be entered on the MCZ1/25 terminal and executed.

MOVE <destination address> <source address> <number>

Moves the contents of a block of memory from the source address to the destination address. The number of bytes specifies the size of the block of memory moved.

NEXT [<number>]

Causes the execution of the next machine instruction, starting at the current value of the PC, and displays all registers after each execution. If the number is not given, 1 is assumed.

REGISTER [<register name>]

Permits the contents of the indicated register to be examined and modified. If no register name is given, all registers will be displayed on one line. If a register name is given, the individual register will be displayed. The specified register name will be displayed followed by its contents followed by a space. If it is desired to change the contents of that register, the new contents are typed after the space. A carriage return, either alone or after typing the new contents, will cause the next register to be displayed. The sequence in which the registers are displayed is: A, B, C, D, E, F, H, L, I, A', B', C', D', E', F', H', L', IX, IY, PC, SP.

SET <hex address> <data> <data> <data>
Stores the given data words into sequential memory locations starting at the given address. A carriage return terminates the list.

If a command is not understood by the monitor, a "?" is typed with the prompt at which time a new command may be given. All numbers are hexidecimal integers without leading zeros. If more than four hex digits are entered, the last four will determine the parameter value. The only delimiter permitted in all commands is the space. Further information on the UNID monitors may be found in the Zilog,Inc., publication titled PROM User's Manual and in reference (9).

Appendix F

Circuit Card Wirewrap Lists.

The wirewrap lists used to build the wirewrap circuit cards are contained under seperate cover. The lists are not published.

Appendix G

DELNET User's Manual.

The DELNET User's Manual describing configuration, operation, and maintenance of the DELNET and the UNID is contained under seperate cover. It is not published.

Vita

Charles E. Papp was born on 22 November 1943 in Cleveland, Ohio. He graduated from Warrensville Heights High School in 1963. He attended the University of California, Davis, California, from which he received a Bachelor of Science degree in Electrical Engineering in June 1974. Upon graduation he received a commission in the USAF through Officer Training School at Lackland AFB, Texas. Following graduation from OTS he was assigned to Keesler AFB, Mississippi, from November 1974 through June 1980. He entered the Air Force Institute of Technology in July 1980.

Permanent Address: 2357 Brown Bark Dr.
Dayton, Ohio 45431

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AFIT/GE/EE/81D-46

2. GOVT ACCESSION NO.
AD-A115 671

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
PROTOTYPE DELNET USING THE UNIVERSAL NETWORK INTERFACE DEVICE

5. TYPE OF REPORT & PERIOD COVERED
MS Thesis

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Charles E. Papp, Captain, USAF

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Air Force Institute of Technology AFIT/EN
Wright-Patterson AFB, OH 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
AirForce Institute of Technology AFIT/EN
Wright-Patterson AFB, OH 45433

12. REPORT DATE
December 1981

13. NUMBER OF PAGES
119

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Dean for Research and
Professional Developme t
Air Force Institute of Technology (ATC)
Wright-Patterson AFB, OH 45433

Lynn Wolaver
LYNN E. WOLAVER
Dean for Research and
Professional Development

18. SUPPLEMENTARY NOTES
Approved for public release; IAW AFR 190-17
~~Frederic C. Lynch,~~ Major, USAF
~~Director of Public~~ Affairs

Air Force Institute of Te [illegible] (ATC)
Wright-Patterson AFB, OH 45433

JUN 1982

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Local Area Network
Local Computer Network
Message Processors

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The objective of this investigation was to complete two prototype microcomputer based message processors and continue construction of a local network using these devices. Using a preliminary design and partially constructed hardware, the message processors were built, documented, tested, and operated. The two message processors, known as Universal Network Interface Devices (UNID), were then configured and operated in a prototype local network.

ATE

MED

8